12.1

IBM Db2 Administration Tool for z/OS User's Guide

SC27-8898-04

2024-07-18 edition

This edition applies to IBM® Db2® Administration Tool for z/OS® 12.1 (product number 5655-DT2) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

About this information

This information provides instructions for customizing and using IBM Db2 Administration Tool for z/OS, a Db2 catalog administration tool.

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

- Plan for the installation of Db2 Admin Tool.
- Install and operate Db2 Admin Tool.
- Customize your Db2 Admin Tool environment.
- Administrate IBM Db2 by using Db2 Admin Tool
- Diagnose and recover from Db2 Admin Tool problems.

Users of this information should understand basic Db2 concepts and facilities.

Always check IBM Documentation (IBM Docs) for the most current version of this publication: https://www.ibm.com/docs/en/db2admintool

Chapter 1. Db2 Admin Tool overview

IBM Db2 Administration Tool for z/OS, or Db2 Admin Tool, is an administration product that can greatly increase the productivity of the entire Db2 for z/OS staff (database administrators, system administrators, and application developers). It is one of several IBM tools that can help you manage database administration and the change management processes.

Among other capabilities, Db2 Admin Tool uses dynamic SQL to access the Db2 catalog tables and to present the information in an easy-to-use ISPF interface. For an overview of the features that Db2 Admin Tool offers, see “What does Db2 Admin Tool do?” on page 76.

What's new in Db2 Admin Tool 12.1

IBM Db2 Administration Tool for z/OS (Db2 Admin Tool) 12.1 introduces new features to support Db2 12 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.

Tip: To see videos for some of the latest enhancements in Db2 Admin Tool, visit Db2 Administration Tool for z/OS video play list (Data and AI on IBM Z channel).

New-function APARs for Db2 Admin Tool 12.1

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 Db2 Admin Tool 12.1. It does not include problem fixes or other maintenance APARs. Any APAR numbers that are followed by (GOC) are for Db2 Object Comparison Tool; all other APARs are for Db2 Admin Tool.

Description	APAR	Date
“Support for STATIME_DDF zparm” on page 10	PH61154	2024-05
“Ability to specify the ADBTEP2 MAXE value in CM batch” on page 11	PH60855 PH60856 (GOC)	2024-04
“Support for new REPAIR utility options” on page 11	PH60526	2024-04
“Support for additional RUNSTATS INDEX options” on page 12	PH58315	2023-12
“Ability to specify MAXPARTITIONS when moving tables from multi-table table spaces to UTS” on page 12	PH58158	2023-11
“Ability to run revoke impact reports in batch” on page 13	PH57694	2023-10
“New CM batch options to include foreign key changes when comparing objects” on page 13	PH55583	2023-10
“Support for new LOAD options” on page 13	PH56994 PH56995 (GOC)	2023-09
“New command to update the high-level qualifier for Db2 catalog data sets” on page 14	PH52946	2023-07
“Improved usability for changing table space and index attributes at the object level” on page 15	PH54716	2023-06
“CM batch support for comparing DDL to DDL” on page 17	PH54480	2023-05

Description	APAR	Date
“Masking support for removing a key label” on page 17	PH54152	2023-05
“Support for new Db2 subsystem parameters and values” on page 18	PH54153	2023-05
“Support for displaying index traverse counts and log reading tasks” on page 18	PH52342	2023-04
“Support for FREE PACKAGE PLANMGMTSCOPE(PHASEOUT)” on page 19	PH51372	2023-03
“Support for additional values for the QUERY_ACCEL_OPTIONS subsystem parameter” on page 19	PH52919	2023-03
“Support for additional COPY utility options” on page 19	PH52344 PH52345 (GOC)	2023-02
“Support for REORG LASTLOG” on page 19	PH50651 PH50652 (GOC)	2022-11
“Eliminate unnecessary changes when comparisons involve objects created prior to Db2 12” on page 20	PH49601	2022-11
“Ability to estimate data set extents” on page 21 “Improvements to page set resizing” on page 21 “Usability improvements to space manager” on page 22 “Space estimator REST API” on page 22	PH42549	2022-09
“REORG SHRLEVEL default change to avoid pending changes” on page 22	PH49639	2022-09
“Improvements to inserting and adding partitions” on page 23	PH48016	2022-08
“Ability to limit UNLOAD_ALTERED_TABLES to destructive alters” on page 23	PH48010	2022-07
“Simplified process for propagating multi-target changes to remote subsystems” on page 24	PH47981	2022-07

Description	APAR	Date
“Ability to prevent binds when a CONTOKEN changes but the package version is the same” on page 24	PH42301	2022-03
“Support for additional Db2 12 subsystem parameters” on page 25	PH42322	2021-12
“ALT as an alternate line command for A” on page 25	PH40458	2021-12
“Support for new REORG INDEX utility options” on page 25 “RECOVER support for auxiliary table spaces” on page 26	PH37342	2021-12
“Ability to specify an object scope when migrating objects” on page 26	PH41894	2021-11
“Support for renaming views” on page 26	PH34099	2021-10
“Support for new utility options” on page 27 “Ability to specify a range of partitions when recovering objects” on page 28	PH37341 PH40162 (GOC)	2021-10
“Ability to restore packages after running the collection clean up function” on page 29	PH35204 PH31960	2021-09
“Support for new object attributes in Db2 12” on page 30	PH31556	2021-09
“Support for additional CREATE TABLE options for implicit table spaces” on page 30	PH40076	2021-09
“Support for additional Db2 12 subsystem parameters” on page 31	PH40074	2021-09
“Ability to control whether foreign keys are dropped during a comparison” on page 31	PH39334 PH39653 (GOC)	2021-09

Description	APAR	Date
“PAGESET_PAGENUM support” on page 32	PH37926	2021-08
“New command to display only explicitly granted privileges” on page 32	PH39331	2021-08
“Ability to view replication status and row count for accelerated tables” on page 33	PH38103	2021-08
“Usability improvements for preserving data when altering the NULLS attribute of a column” on page 33	PH36418 PH38600 (GOC)	2021-07
“Support for new utility options” on page 34	PH35853	2021-06
“Usability enhancements for Db2 12 function level 508 support” on page 36	PH36790	2021-06
“Support for specifying a compression algorithm at the object level” on page 37	PH36485	2021-05
“Support for high availability for accelerator-only tables” on page 37	PH36482	2021-05
“Support for tamper-proof audit policies” on page 37	PH36481	2021-05
“Expanded support for Db2 REST services” on page 38	PH36687	2021-05
“New stored procedure that returns the DDL for a single object” on page 38	PH35130	2021-03
“Expanded ability to view Revoke Impact reports” on page 38	PH32457	2021-03
“Improved specification status for source and target objects in comparisons” on page 39	PH31168	2021-02

Description	APAR	Date
“Ability to specify the detectChanges value when loading accelerated tables” on page 39	PH29571	2021-02
“Support for moving tables from multi-table table spaces to UTS” on page 40	PH31554	2021-01
“Support for Db2 REST services” on page 40	PH31558	2021-01
“Parent objects can be included when migrating tables or table spaces or generating SQL” on page 40 “Extended schema support” on page 41	PH30132	2021-01
“Ability to insert a partition when altering a table space” on page 41	PH27084	2021-01
“Ability to choose an accelerator when launching Db2 Table Editor” on page 42	PH31272	2020-12
“Ability to register changes to column masks and row permissions” on page 42	PH31186	2020-11
“Additional commits for generated SQL statements that update catalog statistics” on page 42	PH30515	2020-11
“Improved consistency and usability for line commands” on page 42	PH28560	2020-11
“SET CURRENT APPLICATION COMPATIBILITY statements included in GEN and DDL output” on page 43	PH28566	2020-11
“Ability to refresh the CM change list by clicking a command” on page 43	PH28892	2020-10
“USS path names accepted when regenerating a DBRM” on page 43	PH26043	2020-10

Description	APAR	Date
“Support for Db2 12 function level 507” on page 43 “APPLCOMPAT value is displayed on the main menu” on page 44	PH28647	2020-09
“Panel scrolling support” on page 44	PH28564	2020-08
“CM batch parameter values are included in ADBMSGS” on page 45	PH28563	2020-08
“Support for Db2 12 fast index traversal” on page 45 “Support for the new REGISTER option for UNLOAD and RUNSTATS” on page 45	PH27090 PH28134 (GOC)	2020-08
“Data preserved when altering the NULLS attribute of a column” on page 46	PH24104 PH27716 (GOC)	2020-08
“Ability to specify target function level when migrating objects” on page 46	PH27083	2020-08
“Support for Db2 12 recovery utility options” on page 46	PH24002	2020-08
“Support for long names in comparison summary reports” on page 47	PH27085	2020-07
“Ability to modify existing traces” on page 47	PH27086	2020-07
“Ability to save compare job options” on page 47	PH27087 PH27130 (GOC)	2020-07
“Support for Db2 12 function level 506” on page 48	PH25686 PH27088	2020-07, 2020-10
“Support for Db2 12 function level 505” on page 48	PH24230	2020-06
“Support for objects that you can no longer create in Db2 12 function level 504” on page 48	PH22951 PH27083	2020-06, 2020-08

Description	APAR	Date
“Use of ADMIN_INFO_SYSPARM instead of DSNWZP” on page 49	PH22951	2020-06
“Support for IBM Db2 AI for z/OS (Db2ZAI)” on page 49 “Support for key labels” on page 49 “New primary command to change the application compatibility level” on page 50	PH22698	2020-05
“Support for inserting partitions” on page 50	PH20648	2020-04
“CM batch option to generate JCL from a WSL” on page 50	PH20257	2020-04
“Ability to view a change that supersedes a change” on page 50 “Exclusion of referential constraints when comparing objects with automatic target selection” on page 51 “Conversion of ADB table spaces to UTS” on page 51	PH22548	2020-03
“Ability to view logical partition numbers and row counts for partitions” on page 51	PH23202	2020-03
“Improvements to CM recovery” on page 52	PH21635	2020-03
“Bind Manager functionality added” on page 52	PH18439	2020-03
“Plan names displayed on ADB21K” on page 52	PH16874	2020-03
“ADBTEP2 can automatically retry failed statements” on page 52	PH20650	2020-02

Description	APAR	Date
“New warning when the compare scope is insufficient” on page 53 “Support for CONCENTRATESTMT and APREUSESOURCE bind options” on page 53 “Support for new FREE PACKAGE options” on page 53 “Ability to view package copy information” on page 53	PH19997	2019-12
“Support for INSERT ALGORITHM” on page 54 “Support for LOAD BACKOUT” on page 54 “Support for dynamic plan stability” on page 54	PH16862	2019-10
“Support for the UNLOAD privilege” on page 54 “Support for REORG DROP_PART” on page 55 “CM uses DSNUTILV instead of DSNUTILU” on page 55 “Ability to specify how CM runs utilities” on page 55 “Support for DELETE FETCH FIRST n ROWS ONLY” on page 55 “Support for new SELECT clauses” on page 56 “Reduced resource contention in CM” on page 56 “Support for the COMPRESSRATIO column in SYSTABLESPACE” on page 56 “APPLCOMPAT V12R1M505” on page 56]	PH10086	2019-10
“Support for updated IDAA trace details” on page 56	PH09487	2019-07
“Support for the INVALIDATECACHE and USE PROFILE statistics options” on page 57 “Db2 Admin Tool dynamically finds the DECP settings” on page 57	PH02457	2019-07

Description	APAR	Date
"Improved function level messages" on page 57 "Improvements to ADBMSGS" on page 57	PH12055	2019-06
"Support for Db2 12 continuous delivery" on page 58 "APPLCOMPAT values for function levels" on page 58	PH06164	2019-05
"Invocation exit to set global variables" on page 58	PH09606	2019-05
"Readability improvements to ADBMSGS" on page 58	PH08484	2019-05
"CM can unload tables when analyzing changes" on page 58	PH00552	2019-04
"CM support for partition-level copies" on page 59	PH03675	2019-02
"Bind avoidance" on page 59 "DBRM regeneration" on page 59 "Collection clean-up" on page 59	PH06267	2018-12
"Ability to view object details in batch mode" on page 59 "Ability to validate imported DDL in CM batch" on page 60 "CM batch parameters for readable WSL" on page 60	PI96053	2018-05
"Ability to alter sequence aliases" on page 60	PI92080	2018-03
"Ability to define restart points in a WSL" on page 60	PI61852	2018-03
"Consolidation of messages into one file" on page 60	PI66475	2018-03
"Improvements to CM batch" on page 61	PI74777	2018-03

Description	APAR	Date
“UTILFROM ADDPART” on page 61	PI80511	2018-03
“New messages for CCSID changes” on page 61	PI82330	2018-03

2024 new-function APARs for Db2 Admin Tool 12.1

The following APARs were delivered in 2024. They introduced enhancements to Db2 Admin Tool 12.1 since the General Availability (GA) date.

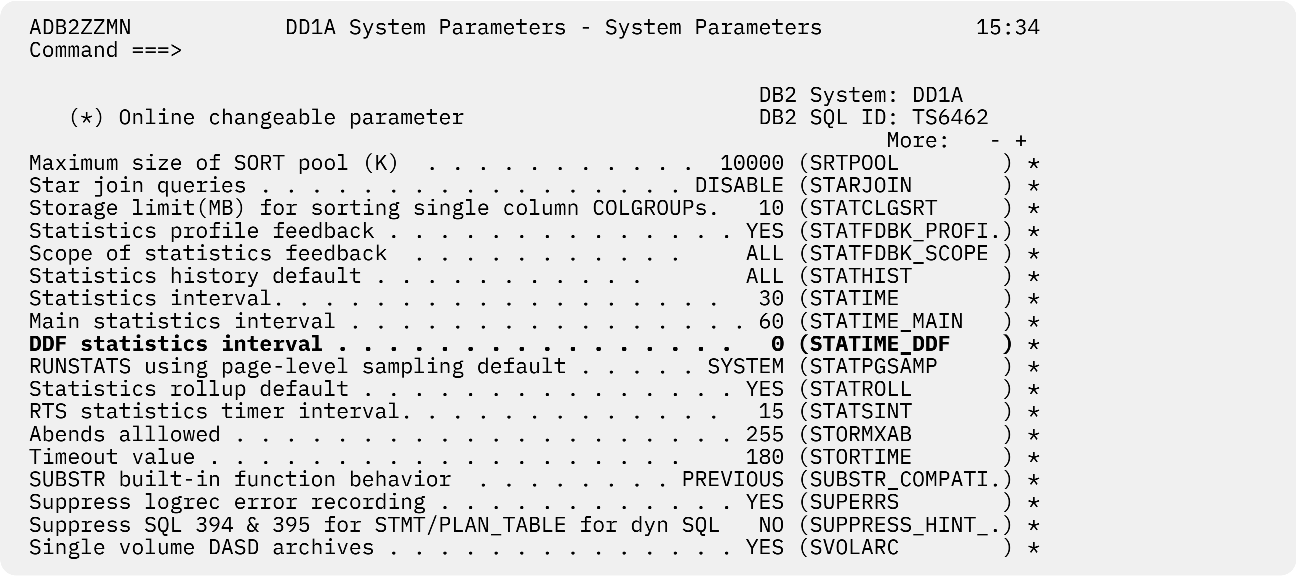
Support for STATIME_DDF zparm

PH61154 - May, 2024

Db2 added the new subsystem parameter (zparm) STATIME_DDF to control the interval of location statistics trace records. This parameter is valid in Db2 12 and Db2 13.

In Db2 Admin Tool, you can now view and update this parameter on the **System Parameters - Tracing and Data Installation (ADB2ZZTR)** panel and the **System Parameters — System Parameters (ADB2ZZMN)** panel:

```
ADB2ZZTR  --- DD1A System Parameters - Tracing and Data Installation -----
Command ===>

                                                         DB2 System: DD1A
                                                         DB2 SQL ID: TS6462
                                                                   More:     +
Start audit trace . . . . . . . . . . . . . . .           > (AUDITST        )
Resident trace table size, 4K multiple . . . . . .          (TRACTBL        )
Start SMF accounting  . . . . . . . . . . . . .           > (SMFACCT        )
Start SMF statistics  . . . . . . . . . . . . .           > (SMFSTAT        )
Statistics interval  . . . . . . . . . . . . . . . . .      (STATIME        ) *
DDF statistics interval  . . . . . . . . . . . . . . .      (STATIME_DDF    ) *
Synchronize statistics recording  . . . . . . . . . . .     (SYNCVAL        ) *
Time between resetting of dataset stats  . . . . . . .      (DSSTIME        ) *
Start monitor trace . . . . . . . . . . . . . .           > (MON            )
Monitor buffer size . . . . . . . . . . . . . . . .         (MONSIZE        )
Include UNICODE information in IFC records  . . . . . .     (UIFCIDS        ) *
Rollup accting aggregation fields . . . . . . . . . .       (ACCUMUID       ) *
System checkpoint frequency (LOGLOAD)  . . . . . .          (CHKFREQ        ) *
UR checkpoint threshold . . . . . . . . . . . . . . . .     (URCHKTH        ) *
Local trace table size, 4K multiplier . . . . . .           (TRACLOC        )
ICF catalog name . . . . . . . . . . . . . . . . .          (CATALOG        ) *
Pseudo close timer  . . . . . . . . . . . . . . . . .       (PCLOSET        ) *
Roll up parallel task's accounting trace  . . . . . . .     (PTASKROL       ) *
```



```
ADB2ZZMN           DD1A System Parameters - System Parameters           15:34
Command ===>

                                                         DB2 System: DD1A
   (*) Online changeable parameter                       DB2 SQL ID: TS6462
                                                                   More:   - +
Maximum size of SORT pool (K)  . . . . . . . . . . .  10000 (SRTPOOL        ) *
Star join queries . . . . . . . . . . . . . . . . . DISABLE (STARJOIN       ) *
Storage limit(MB) for sorting single column COLGROUPs.   10 (STATCLGSRT     ) *
Statistics profile feedback . . . . . . . . . . . . . . YES (STATFDBK_PROFI.) *
Scope of statistics feedback  . . . . . . . . . . .     ALL (STATFDBK_SCOPE ) *
Statistics history default . . . . . . . . . . .        ALL (STATHIST       ) *
Statistics interval. . . . . . . . . . . . . . . . . .   30 (STATIME        ) *
Main statistics interval . . . . . . . . . . . . . . . . 60 (STATIME_MAIN   ) *
DDF statistics interval . . . . . . . . . . . . . . . .   0 (STATIME_DDF    ) *
RUNSTATS using page-level sampling default . . . . . SYSTEM (STATPGSAMP     ) *
Statistics rollup default . . . . . . . . . . . . . . . YES (STATROLL       ) *
RTS statistics timer interval. . . . . . . . . . . . .   15 (STATSINT       ) *
Abends allowed . . . . . . . . . . . . . . . . . . . . . 255 (STORMXAB       ) *
Timeout value . . . . . . . . . . . . . . . . . . . .   180 (STORTIME       ) *
SUBSTR built-in function behavior  . . . . . . . . . PREVIOUS (SUBSTR_COMPATI.) *
Suppress logrec error recording . . . . . . . . . . . . YES (SUPERRS        ) *
Suppress SQL 394 & 395 for STMT/PLAN_TABLE for dyn SQL   NO (SUPPRESS_HINT_.) *
Single volume DASD archives . . . . . . . . . . . . . . YES (SVOLARC        ) *
```

Related information:

“Managing Db2 subsystem parameters” on page 971
PH61154

Ability to specify the ADBTEP2 MAXE value in CM batch

PH60855 (Db2 Admin Tool), PH60856 (Object Comparison Tool) - April, 2024

The batch restart program ADBTEP2 has a parameter, MAXE, that specifies the maximum number of errors (specifically DSN command failures) that can occur before the batch job terminates. You can now also specify this value when using Change Management (CM) batch by using the new CM batch parameter ADBTEP2_MAXE.

Related information:

“ADBTEP2_MAXE” on page 674
PH60855
PH60856

Support for new REPAIR utility options

PH60526 - April, 2024

Db2 Admin Tool has added support for the following options for the Db2 REPAIR utility:

- For table spaces:
  - INSERTVERSIONPAGES
  - LOCATE
  - SET NOAREORPEND, PRO and NOPRO
  - DBD
  - SHRLEVEL CHANGE
- For indexes: LOCATE

You can specify these new options on the **Specify Utility Options - REPAIR TABLESPACE (ADB2USN)** panel and the **Specify Utility Options - REPAIR INDEX (ADB2UXN)** panel.

Additionally, the REPAIR options are simplified on the utilities panels. The **Table Space Utilities (ADB2US)** panel used to list several REPAIR options:

```
  N - Repair nocopypend   NA - Repair nocheckpend    NB - Repair norcvrpend
 NC - Repair catalog                                 NR - Repair noreorgpend
 NW - Repair Auxwarn      NX - Repair Auxcheckpend
```

Similarly, the **Index Utilities (ADB2UX)** panel also listed several REPAIR options:

```
  N - Repair nocopypend      NA - Repair nocheckpend    NB - Repair norcvrpend
 NC - Repair Catalog         NR - Repair norbdpend
 NO - Repair noareorgpend    N* - Repair noareorpendstar
```

Now, only one REPAIR option is listed on each of these panels:

```
  N - Repair
```

When you select this N option, the relevant utility options panel, **Specify Utility Options - REPAIR TABLESPACE (ADB2USN)** panel or **Specify Utility Options - REPAIR INDEX (ADB2UXN)** panel, is automatically displayed so that you can select which REPAIR function you want to run.

Related information:

“Running Db2 utilities on table spaces ” on page 594
“Running Db2 utilities on indexes ” on page 602
Syntax and options of the REPAIR control statement (Db2 12 for z/OS)
PH60526

2023 new-function APARs for Db2 Admin Tool 12.1

The following APARs were delivered in 2023. They introduced enhancements to Db2 Admin Tool 12.1 since the General Availability (GA) date.

Support for additional RUNSTATS INDEX options

PH58315 - December, 2023

Db2 Admin Tool added panel support for the following RUNSTATS INDEX utility options:

- SORTDEVT
- SORTNUM
- INVALIDATECACHE

You can specify these options on the **Specify Utility Options - RUNSTATS (INDEX) (ADB2UXR)** panel.

Related information:

“Running Db2 utilities on indexes ” on page 602
RUNSTATS INDEX syntax and options (Db2 12 for z/OS)
PH58315

Ability to specify MAXPARTITIONS when moving tables from multi-table table spaces to UTS

PH58158- November, 2023

In Db2 Admin Tool, you can move tables from multi-table table spaces to partition-by-growth universal table spaces (UTS) by using the MOVETB command. This command is available as both a line command and primary command on the **Table Spaces (ADB21S)** panel and the **Databases (ADB21D)** panel. When using this comment, you can now specify a maximum number or partitions on the **DB2 Admin Move Table Options (ADB2MVT1)** panel. Previously, the MAXPARTITIONS value was always 1 when using MOVETB.

Related information:

“Moving tables from multi-table table spaces to UTS” on page 503

PH58158

Ability to run revoke impact reports in batch

PH57694 - October, 2023

A revoke impact report helps you determine how authorizations and database objects will be affected by revoking an authorization. Previously, you could request these reports only on the **Revoke *object* Privileges** panels. With this enhancement, you can now request these reports by using a batch job. A sample job is provided in member ADBRIPB in the sample library. Edit this job as needed to add your revoke requests. Then, run it to view the requested revoke impact reports.

Related information:

"Requesting revoke impact reports in batch" on page 517
PH57694

New CM batch options to include foreign key 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 by Db2 Object Comparison Tool.

Related information:

"SOURCE_GEN_FOREIGN_KEYS" on page 718
"TARGET_GEN_FOREIGN_KEYS" on page 723
PH55583

Support for new LOAD options

PH56994 (Db2 Admin Tool), PH56995 (Object Comparison Tool) - September, 2023

Db2 Admin Tool now supports the following LOAD utility options:

- DEFINEAUX
- DRAIN_WAIT
- FLOAT
- FORCE
- FORCEROLLUP
- HISTORY
- INDEXDEFER
- IGNORE
- NOCHECKPEND
- NOSUBS
- OVERRIDE
- PREFORMAT
- RETRY
- RETRY_DELAY

- STATCLGMEMSRT
- SWITCHTIME
- UPDMAXASSIGNEDVAL

You can specify these options on the **Specify Utility Options - LOAD (ADB2UTC)** panel and, when applicable, on the **Table Utilities - LOAD with Cross Loader (ADBPUTLC)** panel

The values for RBALRSN_CONVERSION and SHRLEVEL are also updated to match those values that are currently supported by Db2.

Additionally, the following Change Management (CM) batch options are added for LOAD utility options:

- “UTIL_LOAD_BACKOUT” on page 751
- “UTIL_LOAD_DECFLOAT_ROUNDMODE” on page 752
- “UTIL_LOAD_DEFINEAUX” on page 752
- “UTIL_LOAD_FORCE” on page 753
- “UTIL_LOAD_INDEXDEFER” on page 754
- “UTIL_LOAD_INDEXDEFER_NONUNIQUE” on page 755
- “UTIL_LOAD_NOCHECKPEND” on page 755
- “UTIL_LOAD_NOSUBS” on page 756
- “UTIL_LOAD_PREFORMAT” on page 756
- “UTIL_LOAD_STATISTICS” on page 759
- “UTIL_LOAD_STATISTICS_FORCEROLLUP” on page 759
- “UTIL_LOAD_STATISTICS_HISTORY” on page 760
- “UTIL_LOAD_STATISTICS_INVALIDATECACHE” on page 760
- “UTIL_LOAD_STATISTICS_REPORT” on page 760
- “UTIL_LOAD_STATISTICS_TABLE_SAMPLE” on page 761
- “UTIL_LOAD_STATISTICS_UPDATE” on page 761
- “UTIL_LOAD_STATISTICS_USE_PROFILE” on page 762

Related information:

PH56994
PH56995

New command to update the high-level qualifier for Db2 catalog data sets

PH52946 - July, 2023

You can use the new CATHLQ primary command to update the high-level qualifier that Db2 Admin Tool is to use during the current session for data sets for the Db2 catalog. When you issue this command, the following panel is displayed so that you can easily update or correct the high-level qualifier as needed:

```
ADB2MSPC                  Db2 Space Manager
Command ===>

The Db2 Space Manager is collecting VSAM information for one or
more DB2 catalog page sets. Specify the high-level qualifier
for the Db2 catalog that you are copying. Generally, this value
is the Db2 subsystem ID.  You can also use CATHLQ command to
update the high-level qualifier.

Db2 catalog high-level qualifier . .
```

You can issue CATHLQ from any panel.

Related information:

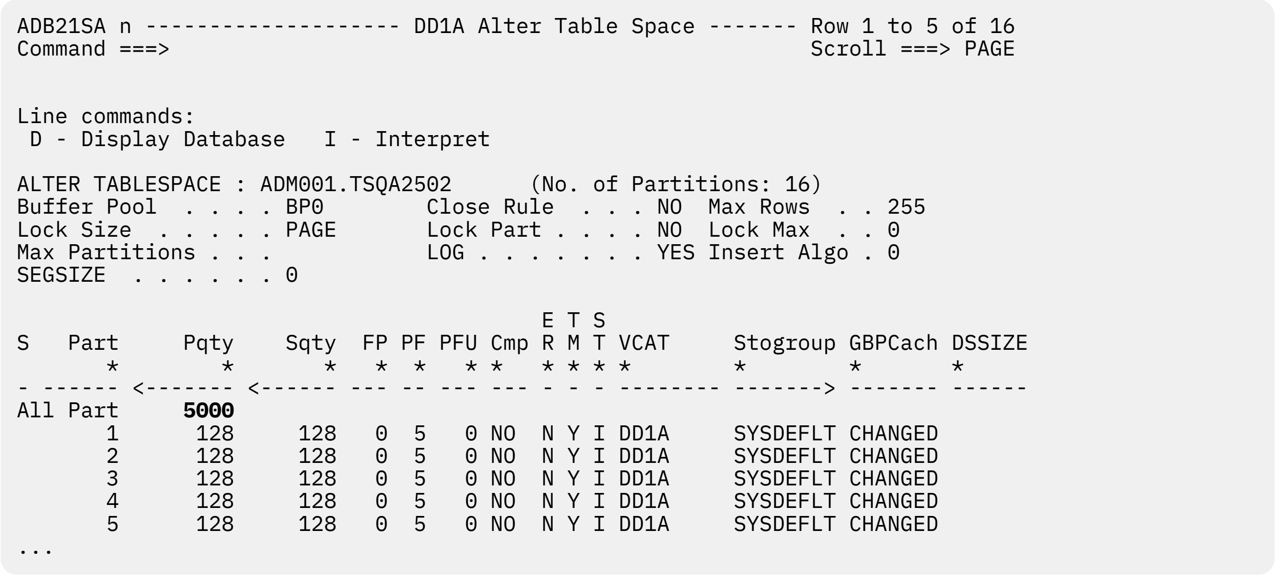
CATHLQ
PH52946

Improved usability for changing table space and index attributes at the object level

PH54716 - June, 2023

When you create or alter table spaces or indexes, you can specify object attributes on the related panels. For partitioned objects, the process of specifying or changing attributes at the object level is now simplified with the following enhancements:

- When you use the AL line command to change certain attributes at the table space or index level, that change can now be done in one ALTER statement. Previously, the attribute value was changed for each partition. For example, suppose you want to change the PQTY attribute of a partitioned table space and therefore, enter the following value on the first row of this panel:



```
ADB21SA n -------------------- DD1A Alter Table Space ------- Row 1 to 5 of 16
Command ===>                                                  Scroll ===> PAGE


Line commands:
 D - Display Database   I - Interpret

ALTER TABLESPACE : ADM001.TSQA2502      (No. of Partitions: 16)
Buffer Pool  . . . . BP0        Close Rule  . . . NO  Max Rows  . . 255
Lock Size  . . . . . PAGE       Lock Part . . . . NO  Lock Max  . . 0
Max Partitions . . .            LOG . . . . . . . YES Insert Algo . 0
SEGSIZE  . . . . . . 0

                                         E T S
S   Part     Pqty    Sqty  FP PF PFU Cmp R M T VCAT     Stogroup GBPCach DSSIZE
       *        *       *   *  *   * *   * * * *        *        *       *
- ------ <------- <------ --- -- --- --- - - - -------- -------> ------- ------
All Part     5000
       1      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
       2      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
       3      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
       4      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
       5      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
...
```

*Figure 1. **Alter Table Space (ADB21SA)** panel*

Previously, this specification resulted in an ALTER TABLESPACE PART statement for each partition of the table space. With this APAR applied, the same panel specification results in only one ALTER TABLESPACE statement that changes this attribute for the entire table space.

This AL enhancement applies to the following attributes:

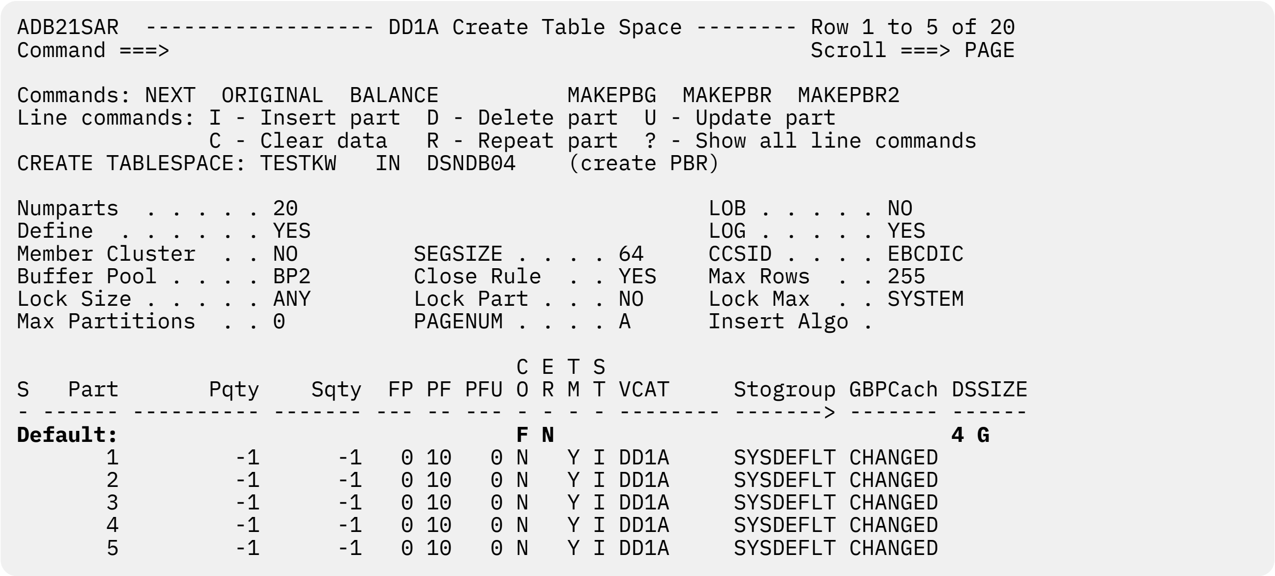
Table space attributes [on the Alter Table Space (ADB21SA) panel]

- PQTY
- SQTY
- FREEPAGE
- PCTFREE
- PCTFREE FOR UPDATE
- COMPRESS
- ERASE
- TRACKMOD
- STORTYPE
- VCATNAME
- STORNAME
- GBPCACHE

Index attributes [on the Alter Index (ADB21XA) panel]

- PQTY
- SQTY
- FREEPAGE
- PCTFREE
- ERASE
- VCATNAME
- STORNAME
- GBPCACHE

- When you create a partitioned table space or index or use the ALT line command to change one of these objects, the DEFAULT row on the resulting panel now displays the object-level attribute values. If no object-level attribute was specified, the value is blank, as shown in the following example:



```
ADB21SAR  ------------------ DD1A Create Table Space -------- Row 1 to 5 of 20
Command ===>                                                  Scroll ===> PAGE

Commands: NEXT  ORIGINAL  BALANCE          MAKEPBG  MAKEPBR  MAKEPBR2
Line commands: I - Insert part  D - Delete part  U - Update part
               C - Clear data   R - Repeat part  ? - Show all line commands
CREATE TABLESPACE: TESTKW   IN  DSNDB04    (create PBR)

Numparts  . . . . . 20                               LOB . . . . . NO
Define  . . . . . . YES                              LOG . . . . . YES
Member Cluster  . . NO         SEGSIZE . . . . 64     CCSID . . . . EBCDIC
Buffer Pool . . . . BP2        Close Rule  . . YES    Max Rows  . . 255
Lock Size . . . . . ANY        Lock Part . . . NO     Lock Max  . . SYSTEM
Max Partitions  . . 0          PAGENUM . . . . A      Insert Algo .

                                       C E T S
S   Part       Pqty    Sqty  FP PF PFU O R M T VCAT     Stogroup GBPCach DSSIZE
- ------ ---------- ------- --- -- --- - - - - -------- -------> ------- ------
Default:                               F N                               4 G
       1         -1      -1   0 10   0 N   Y I DD1A     SYSDEFLT CHANGED
       2         -1      -1   0 10   0 N   Y I DD1A     SYSDEFLT CHANGED
       3         -1      -1   0 10   0 N   Y I DD1A     SYSDEFLT CHANGED
       4         -1      -1   0 10   0 N   Y I DD1A     SYSDEFLT CHANGED
       5         -1      -1   0 10   0 N   Y I DD1A     SYSDEFLT CHANGED
```

*Figure 2. Revised **Create Table Space (ADB21SAR)** panel*

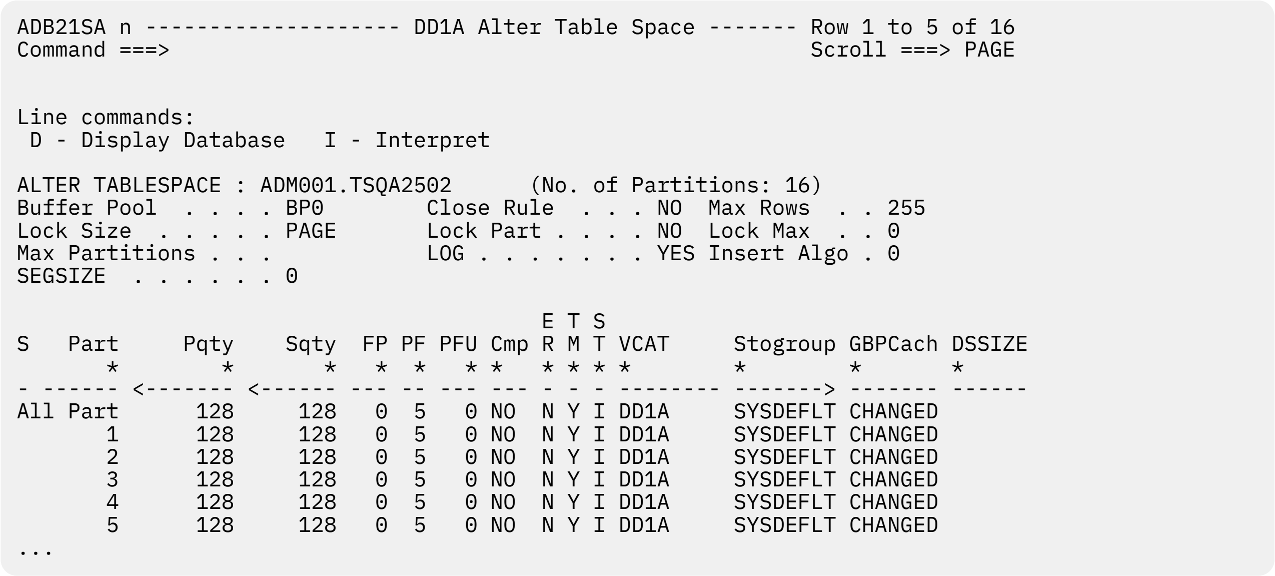
Previously, this DEFAULT row displayed the values for the first partition. This enhancement affects the following panels:

- **Create Table Space (ADB21SAR)** panel
- **Redefine Table Space (ADB21SAR)** panel
- **Create Index - Space (ADB21XAS)** panel
- **Redefine Index - Space (ADB21XAS)** panel

- When you use ALT to change table spaces or indexes, any null attribute values are now displayed as blanks. These null values were previously displayed as question mark characters (?). For example, if an object was created prior to Db2 12, some attributes at the object level might be null. Such values are now displayed as blanks.
- The **Compress** field is removed from the following table space panels, because it is redundant with the **Cmp** or **CO** (Compress) column on these panels:

- **Alter Table Space (ADB21SA)** panel
- **Redefine Table Space (ADB21SAR)** panel
- **Create Table Space (ADB21SAR)** panel

For example, in the following panel, notice that the **COMPRESS** field that was below the **SEGSIZE** field has been removed:



```
ADB21SA n -------------------- DD1A Alter Table Space ------- Row 1 to 5 of 16
Command ===>                                                  Scroll ===> PAGE


Line commands:
 D - Display Database   I - Interpret

ALTER TABLESPACE : ADM001.TSQA2502      (No. of Partitions: 16)
Buffer Pool  . . . . BP0        Close Rule  . . . NO  Max Rows  . . 255
Lock Size  . . . . . PAGE       Lock Part . . . . NO  Lock Max  . . 0
Max Partitions . . .            LOG . . . . . . . YES Insert Algo . 0
SEGSIZE  . . . . . . 0

                                         E T S
S   Part     Pqty    Sqty  FP PF PFU Cmp R M T VCAT     Stogroup GBPCach DSSIZE
       *        *       *   *  *   * *   * * * *        *        *       *
- ------ <------- <------ --- -- --- --- - - - -------- -------> ------- ------
All Part      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
       1      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
       2      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
       3      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
       4      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
       5      128     128   0  5   0 NO  N Y I DD1A     SYSDEFLT CHANGED
...
```

*Figure 3. Revised **Alter Table Space (ADB21SA)** panel*

Related information:

“Altering table spaces” on page 452
PH54716

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 Db2 Object Comparison Tool panels.

To do this comparison, specify the CM batch parameters SOURCE_TYPE='DDL' and TARGET_TYPE="DDL'. (DDL is a new value for TARGET_TYPE.) To specify the data sets that contain the DDL, use either the compare input files (SRCIN DD and TGTIN DD) or the relevant CM batch parameters for the data set names (SOURCE_DSN and TARGET_DSN) .

Related information:

“TARGET_TYPE” on page 725
PH54480

Masking support for removing a key label

PH54152 - May, 2023

You can use the existing masks SGKEYLABL and TBKEYLABL to overwrite the key label for a storage group and table, 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 definitions” on page 274
“Creating masks in a data set” on page 309
PH54152

Support for new Db2 subsystem parameters and values

PH54153 - May, 2023

Db2 Admin Tool added support for the following Db2 subsystem parameters (zparms):

- MAX_UDF
- LA_SINGLESEL_ISOCS_CDY

You can view and change these subsystem parameters on the **System Parameters — System Parameters (ADB2ZZMN)** panel.

Additionally, support is added for the following new parameter values in Db2 12:

Subsystem parameter	New values supported
DDF_COMPATIBILITY	RESET_IFCID402 DISABLE_IMPCAST_NJV IDNTFY_V12_PRIOR_VER IGNORE_TZ
REALSTORAGE_MANAGEMENT	AUTO1

Note: Db2 removed support for DDF_COMPATIBILITY and REALSTORAGE_MANAGEMENT in Db2 13.

Related information:

“Managing Db2 subsystem parameters” on page 971
PH54153

Support for displaying index traverse counts and log reading tasks

PH52342 - April, 2023

In Db2, you can display index traverse counts and log reading tasks by using the DISPLAY STATS command with the appropriate option: INDEXTRAVERSECOUNT or LOGREADERTASKS Db2 Admin Tool now supports both of these options as follows:

- To view the index traverse count for one or more indexes, navigate to the existing **Manage Fast Index Traversal (ADB2Z2I)** panel (option `Z.2I` from the main panel) and select the new option **3 - Display index traverse count**. On the resulting **Display Index Traverse Count (ADB2Z2I3)** panel, you can enter any additional parameters before Db2 Admin Tool issues the command `-DIS STATS(INDEXTRAVERSECOUNT)`.

  Alternatively, you can also view the index traverse count for a database, index, or index partition by using the new DISITIC line command on any of the following panels:

  - **Databases (ADB21D)** panel
  - **Indexes (ADB21X)** panel
  - **Index Parts (ADB21XP)** panel
- To view the log reading tasks, navigate to the **System Administration (ADB2Z)** panel (option Z from the main menu) and select the new option **LT - Display log reader tasks**. On the resulting **Display Log Reading Tasks (ADB2ZLT)** panel, you can enter any additional parameters, such as SCOPE(GROUP), before Db2 Admin Tool issues the command `-DIS STATS(LOGREADERTASKS)`.

Db2 Admin Tool also added support to limit the number of messages that are displayed when you display statistics for fast index traversal. To limit these messages, use the new **LIMIT** option on the **Manage Fast Index Traversal (ADB2Z2I)** panel.

Related information:

“Viewing index traverse counts” on page 978
“Displaying log reading tasks” on page 1001

PH52342

Support for FREE PACKAGE PLANMGMTSCOPE(PHASEOUT)

PH51372 - March, 2023

When you free a package in Db2 Admin Tool, you can now specify the PHASEOUT value for the PLANMGMTSCOPE option on the **Free Package (ADB21KF)** panel if you are running Db2 12 function level 505 or higher.

Related information:

"Freeing packages" on page 919
PH51372

Support for additional values for the QUERY_ACCEL_OPTIONS subsystem parameter

PH52919 - March, 2023

In Db2 Admin Tool, you can now specify the following values for the QUERY_ACCEL_OPTIONS subsystem parameter (zparm):

```
8, 9, 10, 11, and 13
```

You can specify these values on the **System Parameters — System Parameters (ADB2ZZMN)** panel and the **System Parameters - Application Programming (ADB2ZZAP)** panel.

Related information:

"The System Parameters — System Parameters (ADB2ZZMN) panel" on page 973
ACCELERATION OPTIONS field (QUERY_ACCEL_OPTIONS subsystem parameter) (Db2 12 for z/OS documentation)
PH52919

Support for additional COPY utility options

PH52344 (Db2 Admin Tool), PH52345 (Object Comparison Tool) - February, 2023

Db2 Admin Tool added support for the COPY options NOCHECKPAGE and DSNUM. NOCHECKPAGE indicates that COPY is to perform only basic checks on each page. DSNUM identifies the partition or data set to be copied.

You can specify these options on the **Specify Utility Options - COPY (ADB2USC)** panel (for table spaces) and the **Specify Utility Options - COPY INDEX (ADB2UXC)** panel (for indexes). The DSNUM field can accept a range of partitions.

If you are using Change Management (CM) batch, you can specify the NOCHECKPAGE option by using the new the UTIL_COPY_NOCHECKPAGE parameter.

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)
"UTIL_COPY_NOCHECKPAGE" on page 750
PH52344
PH52345

2022 new-function APARs for Db2 Admin Tool 12.1

The following APARs were delivered in 2022. They introduced enhancements to Db2 Admin Tool 12.1 since the General Availability (GA) date.

Support for REORG LASTLOG

PH50651 (Db2 Admin Tool), PH50652 (Object Comparison Tool) - November, 2022

Db2 recently introduced the LASTLOG option for the REORG TABLESPACE and REORG INDEX utilities to help reduce the outage window during REORG SHRLEVEL CHANGE executions. LASTLOG controls whether the utility applies log records during the last log iteration in the LOG phase. This option is valid in both Db2 12 and Db2 13.

APARs PH50651 and PH50652 provide support in Db2 Admin Tool and Object Comparison Tool for this new utility option. You can specify a LASTLOG value on the following REORG option panels:

- **Specify Utility Options - REORG TABLESPACE (ADB2USO1)** panel
- **Specify Utility Options - REORG INDEX (ADB2UXO1)** panel

These panels are displayed only after you specify C (CHANGE) for **SHRLEVEL** on the **Specify Utility Options - REORG (ADB2USO)** panel or the **Specify Utility Options - REORG INDEX (ADB2UXO)** panel

Additionally, you can specify the LASTLOG option by using the following new Change Management (CM) batch parameters:

- UTIL_REORG_LASTLOG
- UTIL_REORG_INDEX_LASTLOG

Note that you must also specify the SHRLEVEL CHANGE option (either `UTIL_REORG_INDEX_ SHRLEVEL = 'C'` or `UTIL_REORG_SHRLEVEL = 'C'`) for LASTLOG to be valid.

The new CM batch parameter UTIL_REORG_INDEX_DRAIN is also added so that you can specify a drain value for REORG INDEX.

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)
Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)
“Running Db2 utilities from Db2 Admin Tool” on page 594
“UTIL_REORG_INDEX_LASTLOG” on page 768
“UTIL_REORG_LASTLOG” on page 771
PH50651
PH50652

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:

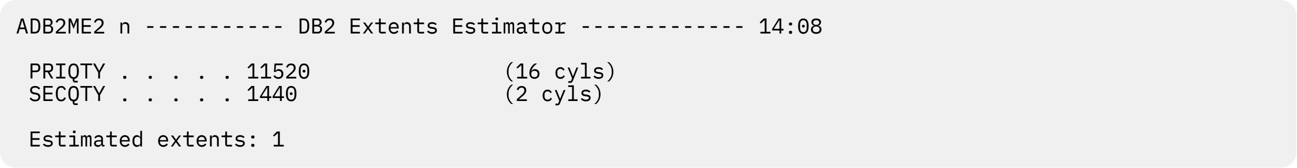
PH49601

Ability to estimate data set extents

PH42549 - September, 2022

The space estimator feature of Db2 Admin Tool is enhanced so that it can now estimate the number of data set extents required based on the primary and secondary space allocations. This enhancement applies to space estimates for table spaces, table space partitions, index spaces, and index partitions.

To view these extent estimates, use the new EXTENTS command on the **DB2 Table Space Estimator (ADB2MES)** panel and the **DB2 Index Space Estimator (ADB2MEX)** panel. When you issue this command, a pop-up window displays the estimated number of extents:



```
ADB2ME2 n ----------- DB2 Extents Estimator ------------- 14:08

 PRIQTY . . . . . 11520               (16 cyls)
 SECQTY . . . . . 1440                (2 cyls)

 Estimated extents: 1
```

You can override the **PRIQTY** and **SECQTY** field values (in KB) on this window to see how the estimated number of extents changes.

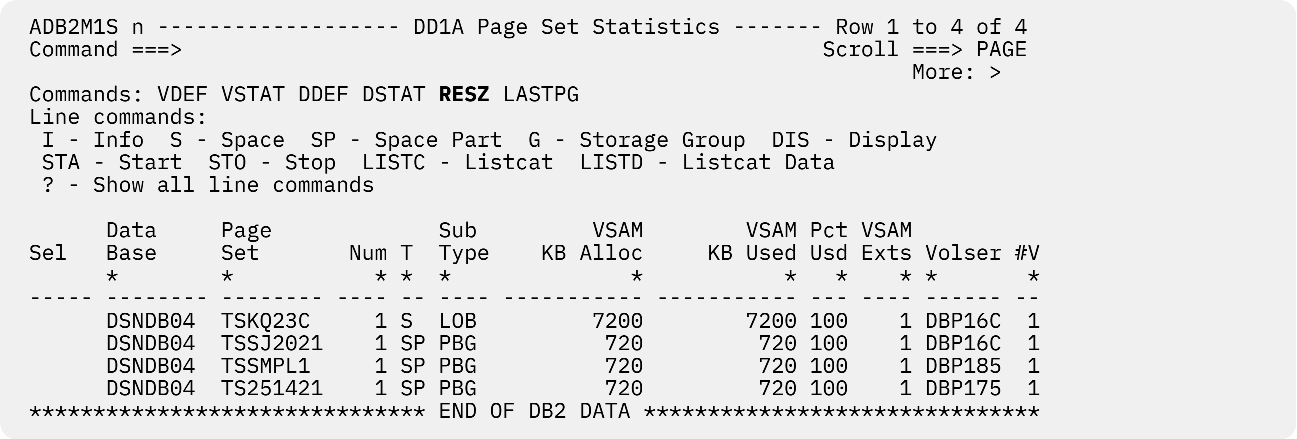
Related information:

“Estimating space requirements for table spaces” on page 635
“Estimating space requirements for index spaces” on page 636
PH42549

Improvements to page set resizing

PH42549 - September, 2022

An existing feature in Db2 Admin Tool is the ability to resize page sets by using the RESZ command in space manager [option `SM.1` from the main **DB2 Administration Menu (ADB2)** panel]:



```
ADB2M1S n ------------------- DD1A Page Set Statistics ------- Row 1 to 4 of 4
Command ===>                                                  Scroll ===> PAGE
                                                                     More: >
Commands: VDEF VSTAT DDEF DSTAT RESZ LASTPG
Line commands:
 I - Info  S - Space  SP - Space Part  G - Storage Group  DIS - Display
 STA - Start  STO - Stop  LISTC - Listcat  LISTD - Listcat Data
 ? - Show all line commands

      Data     Page            Sub         VSAM        VSAM Pct VSAM
Sel   Base     Set        Num T  Type    KB Alloc     KB Used Usd Exts Volser #V
      *        *            * *  *              *           *   *    * *       *
----- -------- -------- ---- -- ---- ----------- ----------- --- ---- ------ --
      DSNDB04  TSKQ23C     1 S  LOB         7200        7200 100    1 DBP16C  1
      DSNDB04  TSSJ2021    1 SP PBG          720         720 100    1 DBP16C  1
      DSNDB04  TSSMPL1     1 SP PBG          720         720 100    1 DBP185  1
      DSNDB04  TS251421    1 SP PBG          720         720 100    1 DBP175  1
******************************* END OF DB2 DATA *******************************
```

Previously, when you specified RESZ, the generated job included STO and STA commands to stop and restart your database. With this APAR, those commands are no longer included in the generated job unless they are required. This change helps minimize unnecessary disruptions to your environment when resizing page sets.

Related information:

“Resizing page sets” on page 633
PH42549

Usability improvements to space manager

PH42549 - September, 2022

You can now access the space manager feature of Db2 Admin Tool directly from the system catalog panels by using the SM and SE line commands. You can use the SM line command to view page set space statistics for databases, table spaces, indexes, and partitions. You can use the SE line command to get space estimates for table spaces, indexes, and partitions.

Additionally, to improve the usability of the **Page Set Statistics (ADB2M1S)** panel, you can now scroll right and left on this panel to view all page set statistics. You can also continue to use the VDEF, VSTAT, DDEF, and DSTAT commands to focus on particular statistics.

Related information:

“Space management” on page 630
“System catalog panels” on page 135
“Displaying page set statistics” on page 631
PH42549

Space estimator REST API

PH42549 - September, 2022

You can now call the space estimator feature of Db2 Admin Tool as a REST API. Three REXX stored procedures are provided for this purpose, one for table space estimates, one for index space estimates, and one for extent estimates. To call space estimator as a REST API, you must first create these stored procedures and then bind them as REST services. You can then invoke them by using REST calls.

Related information:

“Provided REST APIs” on page 876
PH42549

REORG SHRLEVEL default change to avoid pending changes

PH49639 – September, 2022

When using Change Management (CM), Object Comparison Tool, or the ALT command, a change might require a REORG utility operation. In this case, if no value is specified for the REORG SHRLEVEL option, Db2 Admin Tool generates a REORG statement with a default value for SHRLEVEL. With this APAR applied, SHRLEVEL NONE will no longer be generated for these situations, 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; Db2 Admin 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 as follows:

- If you are using CM batch, set `USE_UTILITY_OPTIONS='Y'` and specify a value for `UTIL_REORG_SHRLEVEL`.
- If you are using panels, specify the SHRLEVEL value on the **Specify Utility Options - REORG (ADB2USO)** panel and set one of the following values to YES:
  - **Use Utility Options** on the **Generate Analyze Job (ADB2C11A)** panel (when using CM)
  - **Use customized util opts** on the **Generate Compare Jobs (GOC5)** panel (when using Object Comparison Tool)

If you specify SHRLEVEL NONE and a pending change exists, a warning is issued.

These changes to REORG SHRLEVEL do not apply to other contexts (outside of CM, Object Comparison Tool, and ALT). For example, if you run the REORG utility from the **Table Spaces (ADB21S)** panel, the resulting REORG statements are unchanged.

Related information:

“USE_UTILITY_OPTIONS” on page 729
“UTIL_REORG_SHRLEVEL” on page 777
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

Ability to limit UNLOAD_ALTERED_TABLES to destructive alters

PH48010 – July, 2022

When using Change Management (CM), you can specify whether to unload tables as part of the analyze process by using the CM batch parameter UNLOAD_ALTERED_TABLES or the **Unload altered tables** field on the **Options for Change Functions (ADB2PCO)** panel. Specifying YES ensures that a copy of the data is preserved. However, the unload process can take a significant amount of time and thus increase the overall time it takes to run a change. To help reduce this time, APAR PH48010 introduces a new value for this unload option, DES, which specifies that only those tables with destructive alters are to be unloaded. *Destructive alters* are changes that might result in the loss of data, such as ALTER TABLE DROP COLUMN. DES (or the short value D) is valid for both the CM batch parameter UNLOAD_ALTERED_TABLES and the corresponding **Unload altered tables** field.

Related information:

"UNLOAD_ALTERED_TABLES" on page 727
"Enabling and disabling automatic recreate, reload, or removal of accelerated tables" on page 953
(See panel ADB2PCO)
PH48010

Simplified process for propagating multi-target changes to remote subsystems

PH47981 - July, 2022

The process of analyzing and running multi-target changes on remote subsystems is simplified. You can now analyze and run remote target changes in fewer jobs by specifying the location of the central subsystem and the change name. Use the new CM batch parameter MTC_CENTRAL_LOCATION to specify the location to look for the change.

Previously, when propagating changes to remote subsystems, you had to run a separate analyze job and run job with a unique change name for each target on each remote subsystem. Specifying a central location for the change reduces the number of jobs that you need to run.

For example, assume you create a change with the name MYMTC that affects 30 schemas on your local subsystem DC1A, and you want to propagate this change to 10 remote subsystems. Prior to this enhancement, you had to take the following steps:

1. To implement this change for the 30 schemas on your local susbystem, run two CM batch jobs: one job to analyze the change (`ACTION_ANALYZE_CHANGE='Y'`) and one job to run the change (`ACTION_RUN_CHANGE='Y' `). Both of these jobs specify the change name MYMTC (`CHANGE_NAME='MYMTC'`).
2. For the first remote subsystem, run two jobs (one analyze job and one run job) for each of the 30 target schemas on the subsystem. Each pair of jobs specifies a different change name (for example, `CHANGE_NAME ='MYMTC:TGT1:`*`remote-location-name`*`'`, `CHANGE_NAME='MYMTC:TGT2:`*`remote-location-name`*`'` and so on.) This step has 60 jobs total.
3. Repeat step 2 for the remaining 9 subsystems.

This enhancement simplifies steps 2 and 3. Assuming a DRDA connection exists between the central and remote subsystems, you now need to run only 2 jobs (one analyze job and one run job) on each remote subsystem:

- The analyze job specifies `ACTION_ANALYZE_CHANGE='Y'` , the change name (`CHANGE_NAME='MYMTC'`), and the central location (`MTC_CENTRAL_LOCATION = '`*`dc1a-location`*`'`).
- The run jobs specifies `ACTION_RUN_CHANGE='Y'` , the change name (`CHANGE_NAME='MYMTC'`), and the central location (`MTC_CENTRAL_LOCATION = '`*`dc1a-location`*`'`).

Related information:

"MTC_CENTRAL_LOCATION" on page 705
"Analyzing a multi-target change" on page 801
"Running a multi-target change" on page 806
PH47981

Ability to prevent binds when a CONTOKEN changes but the package version is the same

PH42301 - March, 2022

The bind avoidance program ADBBMA3 is enhanced so that you can specify whether you want a package to be bound if the consistency token (CONTOKEN) has changed but the package version is still the same.

Prior to this enhancement, when you run ADBBMA3 in this situation, the program issues message BND2919I:

```
BND2919I SYSPACKAGE ENTRY=dbrm_name WITH VERSION version HAD CONTOKEN contoken_value
AND DOES NOT MATCH NEW DBRM
```

ADBBMA3 generates the BIND command for the package in the BINDOUT data set and completes normally.

With APAR PH42301, you can control whether you want to bind the package in this scenario. This APAR introduces the new ADBBMA3 option UNIQUE-VERSION, which prevents the DBRM from being bound when the CONTOKEN is changed but the Db2 catalog and DBRM have the same package version. In this scenario, when you specify UNIQUE-VERSION, ADBBMA3 issues a new error message, BDN2920E:

```
BND2920E VERSION version IS NOT UNIQUE FOR PACKAGE package-name IN COLLECTION collid.
SETTING RC=8 BECAUSE OPTION UNIQUE-VERSION WAS SPECIFIED
```

In this case, the BIND command is not generated, and ADBBMA3 ends with RC 8.

Related information:

“Determining whether applications need to be rebound” on page 888
“BND2919I” on page 1222
“BND2920E” on page 1222
PH42301

2021 new-function APARs for Db2 Admin Tool 12.1

The following APARs were delivered in 2021. They introduced enhancements to Db2 Admin Tool 12.1 since the General Availability (GA) date.

Support for additional Db2 12 subsystem parameters

PH42322 - December, 2021

In Db2 Admin Tool, you can now view and update the following Db2 12 subsystem parameter (zparm) values on the **System Parameters — System Parameters (ADB2ZZMN)** panel:

- LOAD_DEL_IMPLICIT_SCALE
- SUBSTR_COMPATIBILITY

Related information:

“The System Parameters — System Parameters (ADB2ZZMN) panel” on page 973
PH42322

ALT as an alternate line command for A

PH40458 - December, 2021

On the **Alter Objects (ADB27CA)** panel, you can use the A line command to alter one of the listed objects. APAR PH40458 enhances this panel so that you can now specify ALT instead of A to alter an object. A and ALT can be used interchangeably.

Related information:

PH40458

Support for new REORG INDEX utility options

PH37342 - December, 2021

You can now specify the following REORG INDEX options on the **Specify Utility Options - REORG INDEX (ADB2UXO)** panel:

- FORCE
- NOSYSUT1
- PARALLEL

Additionally, the values on this panel for RBALRSN_CONVERSION are updated to match those values that are currently supported by Db2.

You can also use the following new Change Management (CM) batch parameters to specify these REORG INDEX options:

- “UTIL_REORG_INDEX_FORCE” on page 767
- “UTIL_REORG_INDEX_NOSYSUT1” on page 768
- “UTIL_REORG_INDEX_PARALLEL” on page 769

Related information:

PH37342

RECOVER support for auxiliary table spaces

PH37342 - December, 2021

Db2 Admin Tool support for the RECOVER utility is enhanced for base table spaces with auxiliary (LOB and XML) table spaces. If you use Db2 Admin Tool to recover an object that has LOB or XML columns, the generated jobs now include RECOVER utility statements for the auxiliary table spaces.

Related information:

PH37342

Ability to specify an object scope when migrating objects

PH41894 - November, 2021

When migrating objects by using the MIG function, you can now specify an object scope. Previously, you had to specify individual objects on an ISPF panel, which was not suitable for large or complex lists of objects.

With this APAR, you can specify the scope of objects that you want to migrate instead of listing them individually. For example, you can use wildcard characters in the scope specification to include many objects; you can also specify objects to exclude from the scope. To specify a scope, take the following actions during the migration process:

1. On the **Migrate Parameters (ADB28M)** panel, set the **Generate MIG jobs in batch** field to YES.

   A new job will be generated and listed with the first group of generated jobs (the jobs that generate the MIG jobs). The name of this job will be either SSTSCBAT or *<Member prefix for combined jobs>*SC, depending on whether you chose to combine job steps.
2. Edit the new job to add an object scope. Add the scope to the SCOPE DD statement in the form of GEN requests. The format and specific requirements for these scope requests are documented in the job comments.

The rest of the migration process is the same as it was previously.

Alternatively, you can continue to specify individual objects on the ISPF panel.

Related information:

“Step 2. Generate the migration batch jobs” on page 522
PH41894

Support for renaming views

PH34099 - October 2021

You can now rename a view by using the REN (Rename) line command on the **Tables, Views, and Aliases (ADB21T)** panel and the **Views (ADB21VV)** panel. The REN line command preserves all authorizations on the view.

Related information:

"Altering views" on page 497
PH34099

Support for new utility options

PH37341 (Db2 Admin Tool), PH40162 (Object Comparison Tool) - October, 2021

Db2 Admin Tool added support for the following Db2 utility options:

Utility	New options on utility panels	New Change Management (CM) batch parameters
LOAD	• PRESORT You can specify this option on the **Specify Utility Options - LOAD (ADB2UTC)** panel and the **Table Utilities - LOAD with Cross Loader (ADBPUTLC)** panel.	• "UTIL_LOAD_PRESORT" on page 756 • "UTIL_LOAD_PRESORTED" on page 757
REBUILD INDEX	• SCOPE values If REBUILD INDEX is run in conjunction with RECOVER, the SCOPE value for REBUILD INDEX on the **Specify Utility Options - REBUILD INDEX (ADB2UXB)** panel is set based on the SCOPE value for RECOVER TABLESPACE on the **Specify Utility Options - RECOVER (ADB2USV)** panel as follows: • When RECOVER TABLESPACE SCOPE is blank or set to UPDATED, REBUILD INDEX SCOPE is set to P. • When RECOVER TABLESPACE SCOPE is set to ALL, REBUILD INDEX SCOPE is set to A.	None
RECOVER	When performing a redirected recovery on objects that have LOB or XML columns, RECOVER utility statements for the AUX table spaces are now added to the resulting job. Previously, you had to edit the JCL to manually add the AUX table spaces.	None
REORG TABLESPACE	• ICLIMIT_DASD • ICLIMIT_TAPE You can specify these options on the **Specify Utility Options - REORG (ADB2USO)** panel.	• "UTIL_REORG_ICLIMIT_DASD" on page 765 • "UTIL_REORG_ICLIMIT_TAPE" on page 766

Related information:

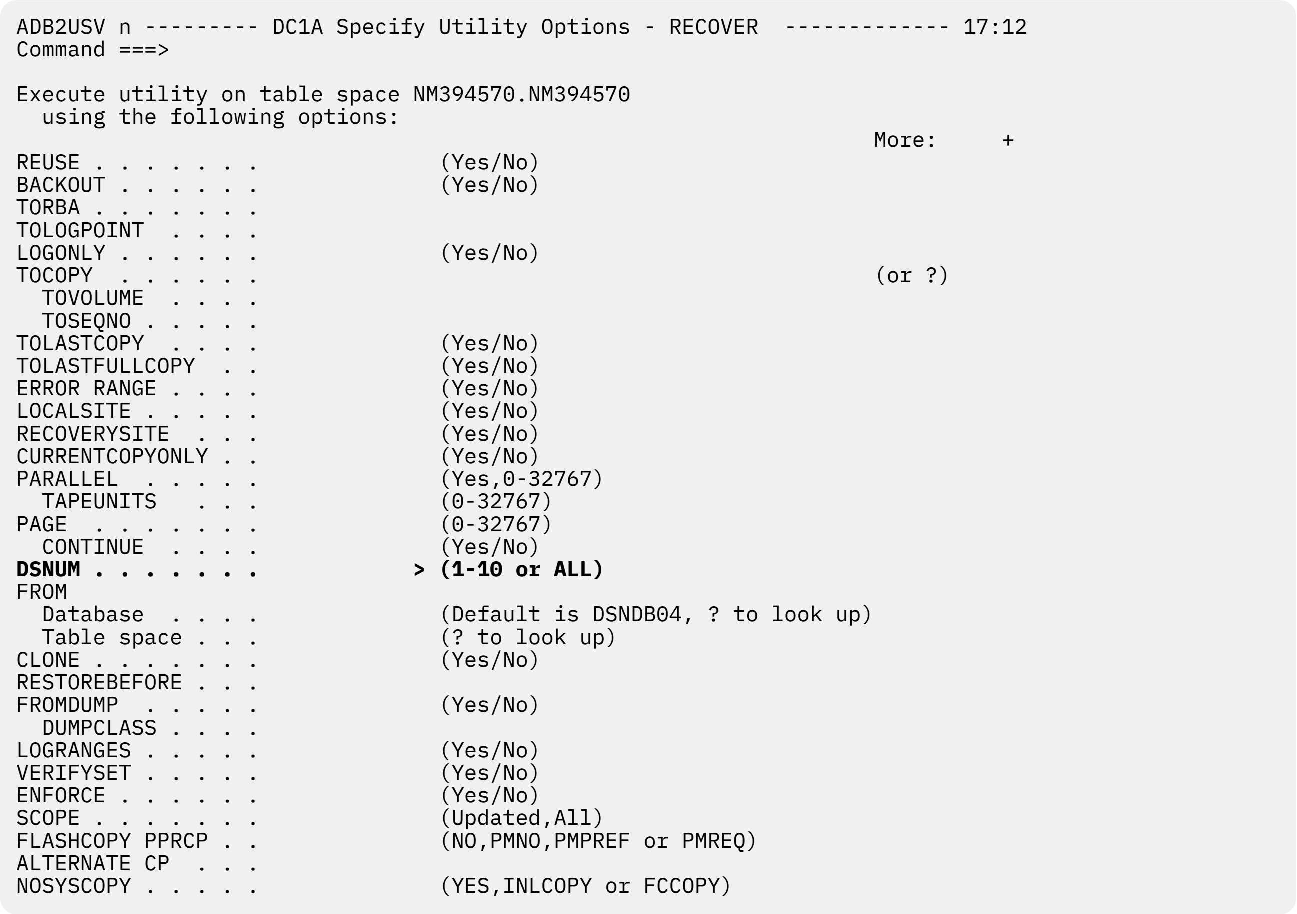
PH37341
PH40162

Ability to specify a range of partitions when recovering objects

PH37341 (Db2 Admin Tool), PH40162 (Object Comparison Tool) - October, 2021

When you use the utility panels in Db2 Admin Tool to recover a table space or index, you can now specify a range of partitions (or data sets for nonpartitioned table spaces). Previously, you could specify only one partition number.

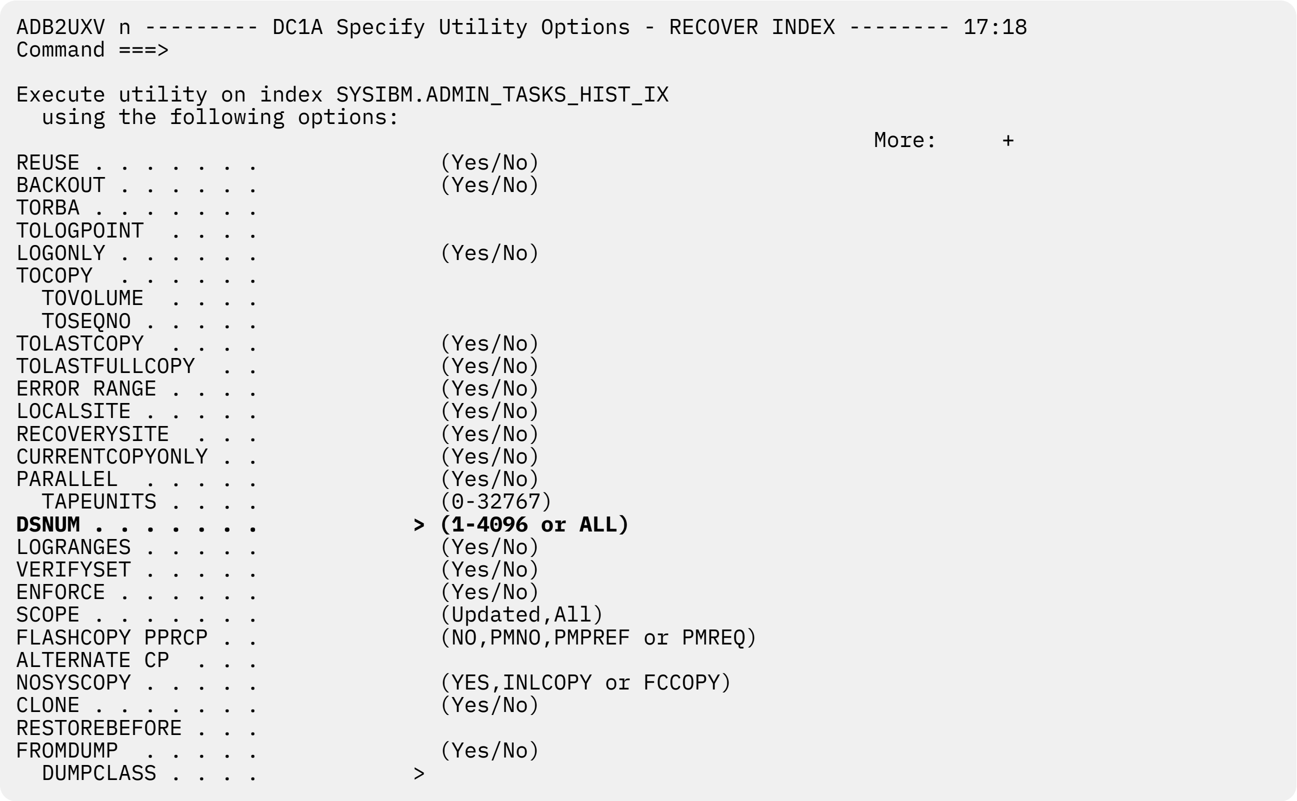
You can specify the partition ranges in the **DSNUM** field on the **Specify Utility Options - RECOVER (ADB2USV)** panel and on the **Specify Utility Options - RECOVER INDEX (ADB2UXV)** panel:



```
ADB2USV n --------- DC1A Specify Utility Options - RECOVER  ------------- 17:12
Command ===>

Execute utility on table space NM394570.NM394570
  using the following options:
                                                                   More:     +
REUSE . . . . . . .              (Yes/No)
BACKOUT . . . . . .              (Yes/No)
TORBA . . . . . . .
TOLOGPOINT  . . . .
LOGONLY . . . . . .              (Yes/No)
TOCOPY  . . . . . .                                                (or ?)
  TOVOLUME  . . . .
  TOSEQNO . . . . .
TOLASTCOPY  . . . .              (Yes/No)
TOLASTFULLCOPY  . .              (Yes/No)
ERROR RANGE . . . .              (Yes/No)
LOCALSITE . . . . .              (Yes/No)
RECOVERYSITE  . . .              (Yes/No)
CURRENTCOPYONLY . .              (Yes/No)
PARALLEL  . . . . .              (Yes,0-32767)
  TAPEUNITS   . . .              (0-32767)
PAGE  . . . . . . .              (0-32767)
  CONTINUE  . . . .              (Yes/No)
DSNUM . . . . . . .            > (1-10 or ALL)
FROM
  Database  . . . .              (Default is DSNDB04, ? to look up)
  Table space . . .              (? to look up)
CLONE . . . . . . .              (Yes/No)
RESTOREBEFORE . . .
FROMDUMP  . . . . .              (Yes/No)
  DUMPCLASS . . . .
LOGRANGES . . . . .              (Yes/No)
VERIFYSET . . . . .              (Yes/No)
ENFORCE . . . . . .              (Yes/No)
SCOPE . . . . . . .              (Updated,All)
FLASHCOPY PPRCP . .              (NO,PMNO,PMPREF or PMREQ)
ALTERNATE CP  . . .
NOSYSCOPY . . . . .              (YES,INLCOPY or FCCOPY)
```

*Figure 4. **Specify Utility Options - RECOVER (ADB2USV)** panel - DSNUM field*

Valid integer values for the **DSNUM** field range from 1 to the number of partitions in the table space. In this example, the **DSNUM** field lists 10 as the maximum number, because the table space has 10 partitions.



```
ADB2UXV n --------- DC1A Specify Utility Options - RECOVER INDEX -------- 17:18
Command ===>

Execute utility on index SYSIBM.ADMIN_TASKS_HIST_IX
  using the following options:
                                                                   More:     +
REUSE . . . . . . .              (Yes/No)
BACKOUT . . . . . .              (Yes/No)
TORBA . . . . . . .
TOLOGPOINT  . . . .
LOGONLY . . . . . .              (Yes/No)
TOCOPY  . . . . . .
  TOVOLUME  . . . .
  TOSEQNO . . . . .
TOLASTCOPY  . . . .              (Yes/No)
TOLASTFULLCOPY  . .              (Yes/No)
ERROR RANGE . . . .              (Yes/No)
LOCALSITE . . . . .              (Yes/No)
RECOVERYSITE  . . .              (Yes/No)
CURRENTCOPYONLY . .              (Yes/No)
PARALLEL  . . . . .              (Yes/No)
  TAPEUNITS . . . .              (0-32767)
DSNUM . . . . . . .            > (1-4096 or ALL)
LOGRANGES . . . . .              (Yes/No)
VERIFYSET . . . . .              (Yes/No)
ENFORCE . . . . . .              (Yes/No)
SCOPE . . . . . . .              (Updated,All)
FLASHCOPY PPRCP . .              (NO,PMNO,PMPREF or PMREQ)
ALTERNATE CP  . . .
NOSYSCOPY . . . . .              (YES,INLCOPY or FCCOPY)
CLONE . . . . . . .              (Yes/No)
RESTOREBEFORE . . .
FROMDUMP  . . . . .              (Yes/No)
  DUMPCLASS . . . .            >
```

*Figure 5. **Specify Utility Options - RECOVER INDEX (ADB2UXV)** panel - DSNUM field*

To specify multiple partitions (or data sets for nonpartitioned table spaces) in the DSNUM field, use dashes to indicate a range and commas to separate each value or range. For example:

The **DSNUM** field is also now scrollable. To enter a longer value, take one of the following actions:

- Place your cursor in the **DSNUM** field and press PF11 to scroll right.
- Type EXPAND in the command field, position your cursor in the **DSNUM** field, and press Enter.

```
DSNUM . . . . . . . 1, 3, 5-7  > (1-4096 or ALL)
```

You can also specify ALL (the default value) to indicate that the entire table space or index space is to be recovered.

Related information:

“Displaying and entering long field values” on page 206
PH37341
PH40162

Ability to restore packages after running the collection clean up function

PH35204, PH31960 - September, 2021

Db2 Admin Tool can now restore one or more packages that were freed by the collection clean up function. The *collection clean up function* runs when you specify the CL line command on the **Collections (ADB21L)** panel.

To restore these packages, use the new CLREST command. You can choose to restore the package with the original DBRM or the DBRM that was regenerated to a backup data set by the clean up function. Db2 Admin Tool copies regenerated DBRMs to PDSs and generates the appropriate BIND commands to bind the DBRMs into packages.

Related information:

"Restoring packages" on page 926
"Db2 Admin Tool primary commands" on page 211
PH35204
PH31960

Support for new object attributes in Db2 12

PH31556 - September, 2021

Db2 Admin Tool supports the following new object attributes in Db2 12:

- For table spaces:
  - COMPRESS
  - FREEPAGE
  - GBPCACHE
  - PQTY
  - SECQTYI
  - STORNAME
  - STORTYPE
  - TRACKMOD
  - VCATNAME
- For indexes:
  - FREEPAGE
  - GBPCACHE
  - PCTFREE
  - PQTY
  - SECQTYI
  - STORNAME
  - STORTYPE
  - VCATNAME

For example, you can generate DDL with these attributes and compare and mask these object attributes.

Note that for Object Comparison Tool, change statements will not be generated to change these target attributes to null if the source was created prior to Db2 12 and thus has null values for these new attributes.

Related information:

"Db2 catalog columns and the corresponding masks" on page 292
PH31556

Support for additional CREATE TABLE options for implicit table spaces

PH40076 - September, 2021

Db2 Admin Tool and Object Comparison Tool now support the following CREATE TABLE options for implicitly created table spaces:

- BUFFERPOOL
- COMPRESS
- LOGGED
- MEMBER CLUSTER

- TRACKMOD

For Db2 subsystems that are running version 12 function level 509 or higher, the options for COMPRESS include **F-Fixedlength** and **H-Huffman**.

For example, the GEN function can generate DDL with these new options. Additionally, when you create a table, you can now specify these options on the **Create Table Options (ADB26TOP)** panel and the **Create Table Options (ADB26CTS)** panel.

Related information:

"Creating tables" on page 334
PH40076

Support for additional Db2 12 subsystem parameters

PH40074 - September, 2021

In Db2 Admin Tool, you can now view and update the following Db2 12 subsystem parameter (zparm) values:

Table 1. Newly supported Db2 12 subsystem parameters

Panel on which you can view and update the parameter	Db2 12 subsystem parameter
System Parameters — System Parameters (ADB2ZZMN) panel	• ALLOW_UPD_DEL_INS_WITH_UR • FTB_NON_UNIQUE_INDEX • INLISTP • LOAD_RO_OBJECTS • STATIME_MAIN • UNION_COLNAME_7 • UTILS_USE_ZSORT • REORG_IC_LIMIT_DASD • REORG_IC_LIMIT_TAPE • REORG_INDEX_NOSYSUT1
System Parameters - Protection and Data Definition (ADB2ZZPR) panel	MFA_AUTHCACHE_UNUSED_TIME

Related information:

"The System Parameters — System Parameters (ADB2ZZMN) panel" on page 973
PH40074

Ability to control whether foreign keys are dropped during a comparison

PH39334 (Db2 Admin Tool), PH39653 (Object Comparison Tool) - September, 2021

You can now specify that you want Object Comparison Tool to drop any foreign keys from target tables that are not also in the corresponding source tables. Previously, if you wanted to drop these foreign keys during a comparison, you had to explicitly add a DROP FOREIGN KEY clause in the ALTER TABLE statement.

To specify this new behavior, use the new field **Drop FKs not in source** on the **Generate Compare Jobs (GOC5)** panel or the new Change Management (CM) batch parameter drop_fks_not_in_source. If you specify YES, foreign keys that are not in the source will be dropped, even if the **Suppress DROP of objects** field (the KEEPTGT parameter in the JCL) is set to YES.

The default value for both the **Drop FKs not in source** field and the drop_fks_not_in_source parameter is NO, which preserves the behavior prior to this APAR.

Related information:

Generating a compare batch job (IBM Db2 Object Comparison Tool for z/OS 12.1.0)
Compare job options (IBM Db2 Object Comparison Tool for z/OS 12.1.0)
“DROP_FKS_NOT_IN_SOURCE” on page 697
PH39334
PH39653

PAGESET_PAGENUM support

PH37926 - August, 2021

The value of the PAGESET_PAGENUM subsystem parameter is used as the default value for PAGENUM when creating partitioned table spaces and tables.

Related information:

PH37926

New command to display only explicitly granted privileges

PH39331 - August, 2021

On authorization panels, you can now filter the authorizations listed so that only explicitly granted authorizations are displayed. To do so, use the new RMIMPL (remove implicit) command. This command removes the rows that represent implicit grants from the list. Implicit grants are grants where **Grantor** is the same as **Grantee** or **GT** (Grantee type) is P.

RMIMPL is available on the following authorization panels:

- **Application Plan Authorizations (ADB2AP)** panel
- **Package Authorizations (ADB2AK)** panel
- **Storage Group Authorizations (ADB2AG)** panel
- **Database Authorizations (ADB2AD)** panel
- **Table Space Authorizations (ADB2AS)** panel
- **Table Authorizations (ADB2AT)** panel
- **Column Authorizations (ADB2AC)** panel
- **Function Authorizations (ADB2AO)** panel
- **Stored Procedure Authorizations (ADB2AO)** panel
- **System Privileges Authorizations (ADB2AZ)** panel
- **Resource Authorizations (ADB2AR)** panel
- **Collection Authorizations (ADB2AL)** panel
- **Schema Authorizations (ADB2AH)** panel
- **Global Variable Authorizations (ADBPAGV)** panel

After using RMIMPL to remove implicit grants, you can add the implicit grants back to the list by using the REFRESH command.

Related information:

“Db2 Admin Tool primary commands” on page 211
PH39331

Ability to view replication status and row count for accelerated tables

PH38103 - August, 2021

You can now view the replication status and row count for accelerated tables directly on the **Display Accelerated Tables (ADBPZAT)** panel. Use the new DISPINFO primary command to display this information in the new columns **R** (for replication status) and **Row Count**:

```
ADBPZAT n ----------- DC1A Display Accelerated Tables ------ Row 1 to 11 of 50
Command ===>                                                Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPREF
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
 ? - Show all line commands

     Table             Table    Server         Row
Sel  Name              Schema   Name     E R A Count    Refresh Time
     *                 *        *        * * * *        *
---- ----------------- -------- -------- - - - -------- ---------------------->
     TBOC5I03          SYSADM   V1       Y N   120      2013-08-21-06.28.00.349
```

To restore the original display, use the REFRESH command or the new DISPREF command. DISPINFO toggles to DISPREF when these new columns are displayed.

The interpretation panel for accelerated tables [**Interpretation of an Object in SYSACCELERATEDTABLES (ADBPZATI)** panel] also now includes fields for replication status and row count. Additionally, the existing **Enabled** field on this panel is changed to **Accel status** for consistency.

Related information:

“Displaying accelerator tables” on page 948
“Db2 Admin Tool primary commands” on page 211
PH38103

Usability improvements for preserving data when altering the NULLS attribute of a column

PH36418 (Db2 Admin Tool), PH38600 (Object Comparison Tool)- July, 2021

This APAR modifies one of the common options for change functions. On the **Options for Change Functions (ADB2PCO)** panel, the **Recreate for NULLS change** option, which was introduced by APAR PH24104, is replaced by the new **Preserve all data** option. Similarly, the corresponding CM batch parameter recreate_for_nulls_change is replaced by the new parameter preserve_all_data.

The new option and parameter have the same meaning as the previous ones. Both the old and new options specify whether the change is to be implemented by alter operations or by dropping and recreating the table. Dropping and recreating the table ensures that the column data is preserved and remains the default behavior.

The difference is that the new option is no longer dependent on another option. Previously, for the **Recreate for NULLS change** option to be applicable, you had to also set the **Unload altered tables** option to YES. With the new option, this requirement is removed. The setting for **Preserve all data** applies regardless of what you specify for **Unload altered tables**. However, if you specify NO (to implement the change by using alter operations), you might also want to set **Unload altered tables** to YES to save a copy of the data. The same dependency is removed for the CM batch parameter. The preserve_all_data parameter applies regardless of what you specify for the unload_altered_tables parameter.

Currently, this option applies only to those changes that modify whether a column can contain null values.

Important: This APAR makes changes to several messages and a CM batch parameter:

- Messages ADB7198 and ADB7385 are changed from informational to warning messages. For details, see “Changed messages” on page 64. If you have any code that relies on these message numbers, change that code to account for the new message suffix.

- The default value of the CM batch parameter util_load_enforce is changed from YES to NO. See "UTIL_LOAD_ENFORCE" on page 753. If that value is not acceptable in any of your CM batch jobs, make changes as needed.

Related information:

Object Comparison Tool: unload altered tables & preserve all data (IBM community: Db2 Tools for z/OS)
"UNLOAD_ALTERED_TABLES" on page 727
"PRESERVE_ALL_DATA" on page 709
"ADB7198W" on page 1139
"ADB7385W" on page 1145
PH36418
PH38600

Support for new utility options

PH35853 - June, 2021

Db2 Admin Tool added support for the following Db2 utility options:

Utility	New utility options on panels	New Change Management (CM) batch parameters
COPYTOCOPY	• FROMLASTFULLCOPY You can now specify FROMLASTCOPY on the **Specify Utility Options - COPYTOCOPY (ADB2US2)** panel and the **Specify Utility Options - COPYTOCOPY INDEX (ADB2UX2)** panel.	None
MODIFY STATISTICS	• DELETE • AGE • DATE You can now run the MODFIY STASTICS utility from Db2 Admin Tool. On the **Table Space Utilities (ADB2US)** panel, specify the new option **MS - Modify statistics**. Then, on the new **Specify Utility Options - MODIFY STATISTICS (ADB2USMS)** panel, specify the MODIFY STATISTICS options that you want to use.	None
REBUILD INDEX	• RBALRSN_CONVERSION values On the **Specify Utility Options - REBUILD INDEX (ADB2UXB)** panel, the values for RBALRSN_CONVERSION are updated to match those values that are currently supported by Db2: • For Db2 11, possible values are B (BASIC), N (NONE), and E (EXTENDED). • For Db2 12, possible values are N (NONE) and E (EXTENDED).	None

Utility	New utility options on panels	New Change Management (CM) batch parameters
RECOVER	- FROM To support redirected recovery, you can now specify **VF - Redirected Recovery** on the **Table Space Utilities (ADB2US)** panel for the target table space. You can subsequently specify the source database and table space on the new **Specify SOURCE for redirected recovery (ADB2USVF)** panel. You can either manually specify or look up the source database and table space. (The specified source will be included in FROM clause in the RECOVER statement.) To look up an object, specify a question mark (?) in the appropriate field. The lookup feature automatically fills the values on the ADB2USVF panel based on the object you select. You can also look up source image copies for the **TOCOPY** option on panel ADB2USVF. This feature allows you to browse SYSIBM.SYCOPY for an image copy of the source table space. Alternatively, you can specify **V - Recover** on the **Table Space Utilities (ADB2US)** panel and then specify the source table space in the new **FROM** field on the **Specify Utility Options - RECOVER (ADB2USV)** panel. You can also now use the lookup feature on this panel for both the **TOCOPY** option and the **FROM** option. Additionally, specifying UPDATED for the **SCOPE** option on this panel now automatically sets the value PENDING for the **SCOPE** option on the subsequent **Specify Utility Options - REBUILD INDEX (ADB2UXB)** panel. For more information about redirected recovery in Db2, see Redirected recovery (Db2 12 for z/OS documentation)	None
REORG TABLESPACE	- STATCLGMEMSRT - FORCE - NOCHECKPEND - SORTNPSI You can now specify these options on the **Specify Utility Options - REORG (ADB2USO)** panel.	- "UTIL_REORG_FORCE" on page 765 - "UTIL_REORG_NOCHECKPEND" on page 774 - "UTIL_REORG_SORTNPSI" on page 778

Utility	New utility options on panels	New Change Management (CM) batch parameters
RUNSTATS	• STATCLGMEMSRT • TABLESAMPLE SYSTEM NONE • SORTDEVT • SORTNUM You can now specify STATCLGMEMSRT on the **Specify Utility Options - RUNSTATS (ADB2USR)** panel. Also on this panel, the existing TABLESAMPLE SYSTEM option has the new valid value NONE. NONE is valid for Db2 12 or later. Additionally, you can now specify SORTDEVT and SORTNUM on the **Specify Utility Options - RUNSTATS (ADB2USRA)** panel. (This RUNSTATS panel is displayed when you invoke utility options during the process of altering or comparing objects.)	• "UTIL_RUNSTATS_PROFILE" on page 784 • "UTIL_RUNSTATS_SORTDEVT" on page 785 • "UTIL_RUNSTATS_SORTNUM" on page 785
UNLOAD	• FROMSEQNO You can now specify FROMSEQNO on the **Specify Utility Options - UNLOAD (ADB2USU)** panel.	None

Related information:

PH35853

Usability enhancements for Db2 12 function level 508 support

PH36790 - June, 2021

When moving tables from multi-table table spaces to partition-by-growth universal table spaces (UTS), the maximum number of tables that you can specify per REORG has changed from 128 to 500. This change ensures that the ADBTEP2 program does not hang due to lock contention.

Additionally, when you move tables, you can use the CHKPDC command to check for pending changes or the CHKNTS command to check whether any of the new table spaces already exist. Those commands are enhanced so that if issues are found, one of the following values is listed in the **Err Stat** on the **Move Tables to PBGs (ADB2MVT2)** panel:

D

The new table space name is a duplicate of a new table space in the list.

P

The old table space already has a pending change that prevents additional ALTER operations.

E

The new table space already exists in the Db2 catalog.

Additionally, if the error is left unresolved, the operations for the affected table spaces are commented out in the resulting job. This behavior prevents the job from failing so that tables can still be moved from those table spaces that do not have any unresolved issues.

Related information:

"Moving tables from multi-table table spaces to UTS" on page 503
PH36790

Support for specifying a compression algorithm at the object level

PH36485 - May, 2021

Db2 12 FL 509 Starting with Db2 12 function level 509, you can specify the fixed-length or Huffman compression algorithm at the object level. Db2 Admin Tool supports this new specification. For example, you can perform the following actions in Db2 Admin Tool:

- Specify the compression algorithm when creating a table space.
- Alter the compression algorithm of a table space or a partition by using the ALT command or AL command.
- Mask these compression algorithm values at the table space level. When you specify a TSCOMPRES mask to overwrite the compression attribute of a table space, you can now specify the new values FIXED and HUFFMAN.
- View the compression algorithm that is used for a table space on the **Table Spaces (ADB21S)** panel and the **Interpretation of an Object in SYSTABLESPACE (ADB21SI1)** panel.
- View the compression algorithm that is used for a partition on the **Table Space Parts for table-space (ADB21SP)** panel and the **Interpretation of an Object in SYSTABLEPART (ADB21SPI)** panel.

Related information:

“Creating masks in a data set” on page 309
“Creating table spaces” on page 333
“Altering table spaces” on page 452
“Mask definitions” on page 274
PH36485

Support for high availability for accelerator-only tables

PH36482 - May, 2021

Db2 12 FL 509 Starting with Db2 12 function level 509, you can define an accelerator-only table in more than one accelerator. To do so, you must first define an accelerator group (a location alias). Then, you can create an accelerator-only table in that accelerator group. That table can then use any accelerator in the group to ensure high availability and workload balancing. You can perform all of these actions in Db2 Admin Tool.

Related information:

“Defining an accelerator group” on page 931
“Modifying an accelerator group” on page 932
“Adding accelerator-only tables” on page 947
Video: Db2 Administration Tool: FL 509 support
PH36482

Support for tamper-proof audit policies

PH36481 - May, 2021

Db2 12 FL 509 Db2 12 function level 509 introduces support for tamper-proof audit policies, which cannot be modified or stopped unless the user is authorized to access the Db2 audit policy profile by a z/OS security product that is external to Db2, such as RACF®. You can create and update these policies in Db2 Admin Tool by using the **Manage Audit Policies (ADBPZAP)** panel. To create a tamper-proof audit policy, set the DB2START column to T when you create or update the policy. You can also now specify a DB2START value of S in Db2 Admin Tool. This value indicates that the audit policy will be started automatically during Db2 startup and can only be stopped by a user with SECADM authority.

Related information:

“Managing audit policies” on page 914
Video: Db2 Administration Tool: FL 509 support
PH36481

Expanded support for Db2 REST services

PH36687 - May, 2021

The initial Db2 Admin Tool support for Db2 REST services was provided in APAR PH31558. This APAR (PH36687) provides the following additional functionality:

- You can create Db2 REST services by using the new CR and B line commands on the **REST Services (ADB21RS)** panel.
- You can view the SQL in an existing REST service package by using the new SQ line command on the **REST Services (ADB21RS)** panel.
- You can view the REST service that is associated with a package by using the new RS line command on the **Packages (ADB21K)** panel.

Related information:

“Creating a Db2 REST service” on page 873
“Managing Db2 REST services” on page 872
“Option RS. REST Services” on page 172
“Option K. Packages” on page 156
PH36687

New stored procedure that returns the DDL for a single object

PH35130 - March, 2021

You can use the new ADBGDDL stored procedure to get the DDL for a single object from the Db2 catalog. The result set includes DDL for only that object; it does not include the DDL for any related objects.

This stored procedure is valid for any of the following object types: alias, database, data type, function, index, storage group, stored procedure, sequence, sequence alias, synonym, table, trigger, table space, view, global variable, role, trusted context, column mask, and permission.

The core function that ADBGDDL uses to generate the DDL is the GEN function. Therefore, you can also optionally request a GEN report, which is returned in another result set.

Related information:

“ADBGDDL stored procedure” on page 886
PH35130

Expanded ability to view Revoke Impact reports

PH32457 - March, 2021

Users without SYSADM authority can now view the complete Revoke Impact report, regardless of whether they have authority to execute the REVOKE statement. If the user ID does not have authority to execute the REVOKE statement, an informational message is displayed.

Previously, the Revoke Impact report displayed only the REVOKE statements that the user ID had authority to execute.

Related information:

“Revoking system authority from an SQLID” on page 355
PH32457

Improved specification status for source and target objects in comparisons

PH31168 - February, 2021

In Db2 Object Comparison Tool, the **DB2 Object Comparison Tool Menu (GOCMENU)** panel lists the specification status for each step. This APAR changes the status text that is displayed when you specify that the definition of the source or target objects is to be extracted from the Db2 catalog. Previously, this specification status was listed as: `DB2 catalog extract specified`. This APAR changes this status to be more specific and include the object type. For example, if you specify that the source definition is to be extracted for databases from the Db2 catalog, the specification status is listed as follows:

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

                                         Specification Status:
   1  - Specify compare source (new)     Database extract specified
   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
```

Other new statuses that are added by this APAR are:

- No database extract specified
- Database & schema extract specified
- Table space extract specified
- No table space extract specified
- Table space & schema extract specified
- Table extract specified
- No table extract specified
- Table & schema extract specified
- Schema extract specified
- No schema extract specified

Related information:

PH31168

Ability to specify the detectChanges value when loading accelerated tables

PH29571 - February, 2021

When loading accelerated tables from Db2 Admin Tool, you can now specify the value of the detectChanges option for the Db2 Analytics Accelerator (IDAA) stored procedure SYSPROC.ACCEL_LOAD_TABLES. Previously, Db2 Admin Tool always set this value to DATA. To specify a value for detectChanges, use the new **Load accelerated tables DETECTCHANGES** field on the **Options for Change Functions (ADB2PCO)** panel or the new Change Management (CM) batch parameter load_accelerated_tables_detect_changes.

Related information:

“Enabling and disabling automatic recreate, reload, or removal of accelerated tables” on page 953
“ADMIN ACCELERATOR LOAD statement” on page 969
“LOAD_ACCELERATED_TABLES_DETECT_CHANGES” on page 704
PH29571

Support for moving tables from multi-table table spaces to UTS

PH31554 - January, 2021

Db2 12 FL 508 Db2 12 function level 508 provides support for moving tables from deprecated multi-table simple or segmented table spaces to partition-by-growth (PBG) universal table spaces (UTS). You can use Db2 Admin Tool to perform this move operation. By moving data to preferred table spaces, you can take advantage of new Db2 functionality that operates only on UTS.

In Db2 Admin Tool, you can use the system catalog panels to find any multi-table table spaces and then select the ones from which you want to move the tables. To move the tables, use the new MOVETB line command or the new MOVETB primary command. Db2 Admin Tool then generates a job that creates the new PBG table spaces and runs the ALTER TABLESPACE statements with the MOVE TABLE clause, the materializing REORG utility operations, and any requested rebinds.

Related information:

“Moving tables from multi-table table spaces to UTS” on page 503
PH31554

Support for Db2 REST services

PH31558 - January, 2021

You can now use Db2 Admin Tool to manage Db2 REST services. You can view information about these services, start and stop them, and free the corresponding package for a REST service. To do so, select the new RS option on the **System Catalog (ADB21)** panel. On the resulting **REST Services (ADB21RS)** panel, you can use the available line commands to perform these actions.

Related information:

Db2 Administration Tool: initial support for Db2 native REST services (IBM community: Db2 Tools for z/OS)
“Managing Db2 REST services” on page 872
“Option RS. REST Services” on page 172
PH31558

Parent objects can be included when migrating tables or table spaces or generating SQL

PH30132 - January, 2021

When you generate SQL for a table or table space, either as part of migrating these objects or by running the GEN function, you now have the option to also generate the SQL for parent objects. For a table, you can generate the SQL for the parent table space and parent database. For a table space, you can generate the SQL for the parent database.

These new options, **CREATE parent DATABASE** and **CREATE parent TABLESPACE**, are available on the **Generate SQL from DB2 catalog (ADB2GENB)** panel and the **Generate SQL from DB2 catalog (ADBP8MG)** panel.

For migration, these new options allow you to easily include the parent objects in the migration process.

Related information:

Db2 Administration Tool: generate DDL for parent table space and database (IBM community: Db2 Tools for z/OS)

"GEN options" on page 360
"Step 1. Specify the objects and information to migrate" on page 520
PH30132

Extended schema support

PH30132 - January, 2021

Db2 Admin Tool has expanded schema functionality to include tables, indexes, aliases, and views.

Previously, when you viewed schemas in Db2 Admin Tool, you could view only the following objects in that schema: sequences, data types, procedures, functions, and variables. Similarly, when you generated SQL for a schema, you could select only those objects in the schema.

With this enhancement, you can now view tables, indexes, aliases, and views in a schema. The **Schemas (ADB21H)** panel is updated to include this information.

Additionally, when you generate DDL for a schema, you can also now choose to include tables, indexes, aliases, and views in the schema. The **Generate SQL from DB2 catalog (ADB2GENB)** panel is updated to include these options. Notice that the panel ID has changed from ADB2GEN to ADB2GENB. This panel replaces all previous ADB2GEN*x* panels.

Lastly, this enhancement provides an option for you filter Db2 objects based on a schema. The **System Catalog (ADB21)** panel is updated to include a new **Schema** field in the selection criteria.

Related information:

Db2 Administration Tool: improved schema support (IBM community: Db2 Tools for z/OS)

"Option H. Schemas" on page 152
"Generating SQL to re-create a Db2 object (reverse engineering)" on page 357
"The System Catalog (ADB21) panel" on page 135
PH30132

Ability to insert a partition when altering a table space

PH27084 - January, 2021

APAR PH20648 provided support in Db2 Admin Tool for inserting partitions. (See "Support for inserting partitions" on page 50.) This APAR builds on that support by providing another way to insert a partition. To improve usability, you can now also insert a partition when altering a table space. When you alter a table space and increment the NUMPARTS value or issue the VALUES command, the subsequent **Alter Partitioned Table (ADB21TAV)** panel includes an INS line command for inserting a partition and a new ADD command for adding a partition.

Additionally, a new ORIGINAL command is included on the **Alter Partitioned Table (ADB27CPV)** panel and on the **Alter Partitioned Table (ADB21TAV)** panel when that panel is displayed after issuing the AL line command on a table. The ORIGINAL command resets all changes to their original values.

Related information:

Db2 Administration Tool: Insert partition (IBM community: Db2 Tools for z/OS)
"Altering table spaces" on page 452
"Redefining partitions in a partitioned table space that uses table-controlled partitioning" on page 459
PH27084

2020 new-function APARs for Db2 Admin Tool 12.1

The following APARs were delivered in 2020. They introduced enhancements to Db2 Admin Tool 12.1 since the General Availability (GA) date.

Ability to choose an accelerator when launching Db2 Table Editor

PH31272 - December, 2020

When you edit a table by launching Db2 Table Editor from Db2 Admin Tool, you can now choose an accelerator from a list of accelerators that are associated with that table. The selected accelerator or accelerators are then used by Db2 Table Editor.

To enable this function, you must set the **Pass accelerator name to Table Editor** field to YES on the **DB2 Parameters (CCQPDB2)** panel when customizing Db2 Admin Tool.

Related information:

“Defining Db2 parameters” on page 106
“Invoking Db2 Table Editor from Db2 Admin Tool” on page 506
PH31272

Ability to register changes to column masks and row permissions

PH31186 - November, 2020

When using Change Management (CM), you can now register changes to column masks and row permissions. Previously, these changes were restricted in CM.

Specifically, you can now register the following changes:

- CREATE and ALTER statements for masks and permissions
- ALTER TABLE ACTIVATE and DEACTIVATE ROW and COLUMN ACCESS

Additionally, you can also now specify ignores for fields in the SYSCONTROLS catalog table.

Related information:

“Ignore fields” on page 847
PH31186

Additional commits for generated SQL statements that update catalog statistics

PH30515 - November, 2020

To reduce potential catalog lock contention, Db2 Admin Tool now adds COMMIT statements when generating more than 500 SQL statements that update catalog statistics. COMMIT statements are added as needed so that no more than 500 update statements are executed between commits. If necessary, COMMIT statements are added regardless of what you specify in the **Commit statements per** field on the **Generate SQL from DB2 catalog (ADB2GENB)** panel.

Related information:

“Generating SQL to re-create a Db2 object (reverse engineering)” on page 357
PH30515

Improved consistency and usability for line commands

PH28560 - November, 2020

The ? line command is now valid on all table display panels. It displays all valid line commands for the current panel along with their descriptions.

Additionally, the process of defining your own line commands is simplified. All line commands are defined in one table, ADBLCMDS.

Related information:

“Commands in Db2 Admin Tool ” on page 208
“Defining your own line commands and primary commands for specific panels” on page 1042
PH28560

SET CURRENT APPLICATION COMPATIBILITY statements included in GEN and DDL output

PH28566 - November, 2020

When Db2 Admin Tool generates the DDL for a view, synonym, or hash-organized table, the generated output now includes any necessary SET CURRENT APPLICATION COMPATIBILITY statements. For example, to recreate synonyms or hash-organized tables when the current Db2 function level is 504 or higher, the application compatibility function level must first be set to 503 or lower. In this case, the following statement is added to the GEN or DDL output:

```
SET CURRENT APPLICATION COMPATIBILITY = 'V12R1M503'
```

You can generate the DDL for an object by using either the GEN or DDL line commands or the GEN primary command. Additionally, Db2 Admin Tool generates DDL to recreate objects as part of the running other functions, such as object comparisons.

Related information:

“Sample output from generating SQL” on page 385
PH28566

Ability to refresh the CM change list by clicking a command

PH28892 - October, 2020

When using Change Management (CM), you can refresh the list of changes on the **CM - Changes (ADB2C11)** panel by clicking the new REFRESH command on the panel. Any status changes are displayed immediately. You no longer have to type the REFRESH (or REF) command to reload the data on the panel.

Related information:

“Displaying changes” on page 870
PH28892

USS path names accepted when regenerating a DBRM

PH26043 - October, 2020

When using the Bind Manager function to regenerate a DBRM, you can now specify a z/OS UNIX System Services (USS) path name for the output. The package name is appended to the specified path name. Previously, you could specify only a partitioned data set (PDS). Now, you can specify either a PDS or USS path name on the **Regenerate a DBRM Member (ADBPBMRD)** panel. Additionally, on this panel, you can optionally specify a Db2 version and language for the DEFAULTS control statement.

Related information:

“ Regenerating DBRMs” on page 927
PH26043

Support for Db2 12 function level 507

PH28647 - September, 2020

Db2 12 FL 507 Db2 12 function level 507 introduces the new option CREATE OR REPLACE for external procedures and native SQL procedures. This new option provides an easy way for you to change your stored procedure definitions. Db2 Admin Tool supports this new option when performing operations on stored procedures. For example, you can specify that you want to use CREATE OR REPLACE when creating or modifying procedures. Also, Db2 Admin Tool can generate CREATE OR REPLACE PROCEDURE statements when comparing objects or when generating DDL for other functions.

The following panels have changes to support this new CREATE OR REPLACE option:

Panel	Change
Generate SQL from DB2 catalog (ADB2GENB) panel	A new R option for **CREATE PROCEDURE** field
Create Procedure (ADB26CO) panel	A new **Use CREATE OR REPLACE** field

Function level 507 also introduces specific names for procedures with the new SPECIFIC clause. In Db2 Admin Tool, you can use the existing STPNAME mask to mask the specific name for a native SQL procedure.

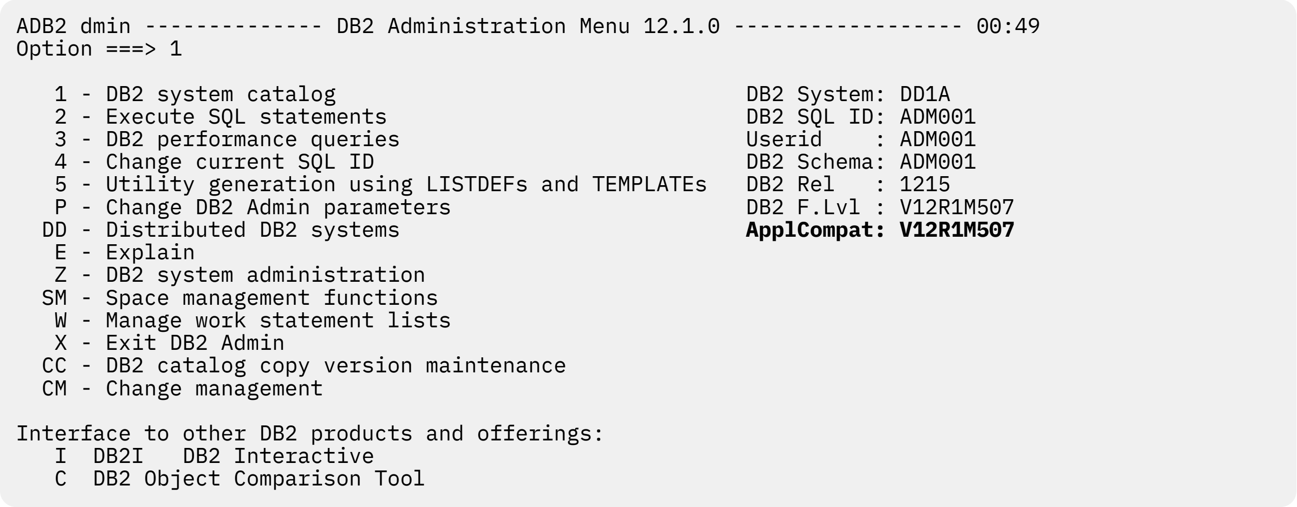
Related information:

“Creating stored procedures” on page 1014
“Scenario: Creating native SQL procedures” on page 1016
“Mask definitions” on page 274
PH28647

APPLCOMPAT value is displayed on the main menu

PH28647 - September, 2020

The **DB2 Administration Menu (ADB2)** panel now displays the function level value of the CURRENT APPLICATION COMPATIBILITY special register:



```
ADB2 dmin -------------- DB2 Administration Menu 12.1.0 ------------------ 00:49
Option ===> 1

   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   : 1215
   P - Change DB2 Admin parameters                    DB2 F.Lvl : V12R1M507
  DD - Distributed DB2 systems                        ApplCompat: V12R1M507
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

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

Related information:

“APPLCOMPAT function level” on page 71
PH28647

Panel scrolling support

PH28564 - August, 2020

The following panels are now scrollable to the right and left to display all columns of the associated Db2 catalog tables:

- **Real-time Statistics for Table (ADBP1RTS)** panel
- **Information from SYSIBM.SYSCOPY (ADB21I)** panel

- **Stored Procedures (ADB21O)** panel
- **Application Plans (ADB21P)** panel
- **Sequence Objects (ADB21Q)** panel
- **Functions (ADB21F)** panel
- **Data Types (ADB21E)** panel
- **Triggers (ADB21J)** panel

Related information:

PH28564

CM batch parameter values are included in ADBMSGS

PH28563 - August, 2020

The consolidated messages file, ADBMSGS, now lists values of CM batch parameters. ADBMSGS can include the following lists of values:

- The initial and profile values for the parameters
- The parameter values that were provided in an invocation override (by using the PARMS DD statement)
- The final parameter values that were used

Each of these lists is identified by message ADB7957I in ADBMSGS. Within each list, the parameters are ordered alphabetically. Parameters with blank values are not listed.

The first list (initial and profile values) is always included in ADBMSGS. The second two lists (invocation override values and final values) are optional. To control whether these last two optional lists are included, use the new CM batch parameter list_options. By default, these optional lists are not included in ADBMSGS.

Related information:

"Consolidating messages into a single file" on page 246
"ADB7957I" on page 1156
"LIST_OPTIONS" on page 704
PH28563

Support for Db2 12 fast index traversal

PH27090 (Db2 Admin Tool), PH28134 (Object Comparison Tool) - August, 2020

You can use Db2 Admin Tool to manage fast index traversal, a new feature of Db2 12. Use the new 2I command on the **System Administration (ADB2Z)** panel to control which indexes use fast index traversal and display memory usage.

Related information:

Chapter 5, "Db2 systems administration," on page 899
"Fast index traversal" on page 976
PH27090
PH28134

Support for the new REGISTER option for UNLOAD and RUNSTATS

PH27090 (Db2 Admin Tool), PH28134 (Object Comparison Tool) - August, 2020

When running the UNLOAD or RUNSTATS utilities in Db2 12, you can use the new REGISTER option in data sharing environments to specify whether pages that are read by the utility are registered with the coupling facility. In Db2 Admin Tool, you can specify the REGISTER value by using the new CM batch parameters util_runstats_shrlevel_change_register and util_unload_shrlevel_change_register or by specifying a value on one of the following panels:

- **Specify Utility Options - RUNSTATS (ADB2USRA)** panel
- **Specify Utility Options - UNLOAD (ADB2USU)** panel

Related information:

“UTIL_RUNSTATS_SHRLEVEL_CHANGE_REGISTER” on page 785
“UTIL_UNLOAD_SHRLEVEL_CHANGE_REGISTER” on page 789
PH27090
PH28134

Data preserved when altering the NULLS attribute of a column

PH24104 (Db2 Admin Tool), PH27716 (Object Comparison Tool) – August, 2020

Important: This APAR is overridden by APAR PH36418, which replaces the **Recreate for NULLS change** field and recreate_for_nulls_change CM batch parameter with the **Preserve all data** field and the perserve_all_data CM batch parameter. See “Usability improvements for preserving data when altering the NULLS attribute of a column” on page 33.

When you alter whether a column can contain null values (the NULLS value in the SYSCOLUMNS catalog table), Db2 Admin Tool now drops and recreates the table by default. Db2 Admin Tool unloads the table, drops the table, recreates the table, and then loads the table. This behavior ensures that the column data is preserved.

You have the option to change this default behavior if you also specify that the altered tables are unloaded when the change is analyzed. (The **Unload altered tables** field on the **Options for Change Functions (ADB2PCO)** panel is set to YES.) In this case, you can change the value of the new field **Recreate for NULLS change** on the **Options for Change Functions (ADB2PCO)** panel to NO. (YES is the default value.) NO indicates that the column is to be altered by using ALTER TABLE statements instead of dropping and recreating the table. However, be aware that when changing the NULLS attribute of a column, specifying **Recreate for NULLS change** = NO might result in data loss. If the **Unload altered tables** field is set to NO, Db2 Admin Tool always drops and recreates the table when altering the NULLS attribute of a column.

This APAR also introduces a new corresponding Change Management (CM) batch parameter, recreate_for_nulls_change, to control this behavior.

Related information:

PH24104
PH27716

Ability to specify target function level when migrating objects

PH27083 - August, 2020

You can now specify the function level for the target system when migrating objects. You can specify this value on the **Generate SQL from DB2 catalog (ADBP8MG)** panel in the new field **Target Function Level**. If the function level for the target system is not specified, the Migrate function uses the function level from the current APPLCOMPAT value on the source system.

Related information:

“Step 1. Specify the objects and information to migrate” on page 520
PH27083

Support for Db2 12 recovery utility options

PH24002 - August, 2020

You can use Db2 Admin Tool panels to specify utility options that are new in Db2 12 for the RECOVER utility and MODIFY RECOVERY utility.

Specifically, you can specify the following RECOVER options on the **Specify Utility Options - RECOVER (ADB2USV)** panel and the **Specify Utility Options - RECOVER INDEX (ADB2UXV)** panel:

- FLASHCOPY_PPRCP
- ALTERNATE_CP
- NOSYSCOPY
- SCOPE (which is part of the TORBA and TOLOGPOINT clauses)

You can also specify the following MODIFY RECOVERY options on the **Specify Utility Options - MODIFY (ADB2USM)** panel:

- DELETEDS
- NOCOPYPEND
- FLASHCOPY ONLY (which is part of the DELETE and RETAIN clauses)

Related information:

PH24002

Support for long names in comparison summary reports

PH27085 - July, 2020

The object comparison summary report is enhanced to display object names that are longer than 18 characters. These long names are displayed across multiple rows, in the same column and with the same indentation.

Related information:

Compare Db2 Objects sample report 1 (IBM Db2 Object Comparison Tool for z/OS 12.1.0)
PH27085

Ability to modify existing traces

PH27086 - July, 2020

In Db2 Admin Tool, you can now use panels to modify Db2 instrumentation traces. On the **Display/ Manage Traces (ADB2Z2T2)** panel, when you issue the MT line command, a new panel is displayed, the **Modify Trace (ADB2Z2TM)** panel. On this panel, you can modify the trace classes, IFCIDs, and comment for the trace. Based on the selected options, Db2 Admin Tool generates the command for modifying the trace and executes it.

Related information:

"Managing traces" on page 905
PH27086

Ability to save compare job options

PH27087 (Db2 Admin Tool), PH27130 (Object Comparison Tool)- July, 2020

Previously, when you saved a dialog in Object Comparison Tool, the options on the **Generate Compare Jobs (GOC5)** panel were not included. With this APAR, all options on the **Generate Compare Jobs (GOC5)** panel and its subordinate panels (those panels that are displayed when you specify options such as UO and CO) can now be saved and restored.

Related information:

Saving dialogs (IBM Db2 Object Comparison Tool for z/OS 12.1.0)
PH27087
PH27130

Support for Db2 12 function level 506

PH25686 - July, 2020, PH27088 - October, 2020

Db2 12 FL 506 Starting with Db2 12 function level 506, you can use alternative built-in function names in Db2 Admin Tool. Additionally, Object Comparison Tool handles the implicit dropping of explicitly created universal and LOB table spaces as follows:

- Object Comparison Tool will not recreate a table space that was implicitly dropped when a table or LOB column was dropped and is not present in the source for the comparison.
- Object Comparison Tool will recreate an implicitly dropped table space if needed to drop and recreate an object.

Related information:

PH25686
PH27088

Support for Db2 12 function level 505

PH24230 - June, 2020

Db2 12 FL 505 Db2 Admin Tool supports the following enhancements in Db2 12 function level 505:

- **The ability to specify DECFLOAT columns in an index and as a key:** In Db2 Admin Tool, you can specify a column of type DECFLOAT as part of an index or primary key on the following panels:
  - **Redefine Index (ADB21XAR)** panel
  - **Alter Index - Add Columns (ADB21XAA)** panel
  - **Create Primary or Unique Key (ADBP7CTP)** panel
  - **Create Primary or Unique Key (ADB26CTP)** panel
- **Rebind phase-in for packages:** You can view the new CopyID value in Db2 Admin Tool on the following panels:
  - The **Package Dependencies (ADB21KD)** panel, which is now scrollable so that you can view all columns in the SYSPACKDEP catalog table.
  - **Interpretation of an Object in SYSPACKDEP (ADB21KI)** panel

Related information:

PH24230

Support for objects that you can no longer create in Db2 12 function level 504

PH22951 - June 2020, PH27083 - August, 2020

Db2 12 FL 504 Starting with Db2 12 function level 504, Db2 does not allow you to create synonyms, segmented or partitioned (non-UTS) table spaces, or hash-organized tables. APAR PH22951 ensures that Db2 Admin Tool can still process these deprecated objects. Therefore, if you have activated function level 504 and have existing objects of one of these deprecated types, you can still use Db2 Admin Tool to do functions such as migrate these objects to another subsystem, generate the SQL for these objects, and use copies of the Db2 catalog.

Because certain functions in Db2 Admin Tool require objects to be recreated, Db2 Admin Tool needs to be able to create the objects that Db2 does not allow you to create in function level 504. This APAR provides that ability. Whether Db2 Admin Tool can recreate these deprecated objects is controlled by new ADBTEP2 parameters:

- For the CM batch interface, this ability is controlled by the new adbtep2_retry_deprecated_obj parameter.
- For the online interface, this ability is controlled by the **Retry Deprecated Obj** field on the **Batch Job Utility Parameters (ADB2UPA)** panel.

By default, these parameters are set to YES.

An operation might result in an SQLCODE -20008 if the current application compatibility level is function level 504 or higher and a deprecated object has to be created. The new ADBTEP2 parameters handle this situation.

Related information:

Video: Db2 Administration Tool: FL 504 support
“ADBTEP2_RETRY_DEPRECATED_OBJ” on page 678
“Batch job parameters for utility jobs” on page 605
“RETRY_DEPRECATED_OBJ” on page 579
PH22951
PH27083

Use of ADMIN_INFO_SYSPARM instead of DSNWZP

PH22951 - June, 2020

Db2 Admin Tool now calls the ADMIN_INFO_SYSPARM stored procedure, instead of the DSNWZP stored procedure, to get Db2 subsystem parameter values. DSNWZP is deprecated in Db2 12. You must ensure that ADMIN_INFO_SYSPARM is operational.

Related information:

Deprecated function in Db2 12 (Db2 12 for z/OS)
ADMIN_INFO_SYSPARM stored procedure (Db2 12 for z/OS)
“Optimizing ADMIN_INFO_SYSPARM and DSNZPARM settings for GEN and DDL” on page 116
“Get DB2 ZPARM” on page 236
PH22951

Support for IBM Db2 AI for z/OS (Db2ZAI)

PH22698 - May, 2020

Db2 12 FL 503 Db2 12 function level 503 introduced support for Db2ZAI, which leverages machine learning technology to improve the Db2 optimizer. You can start, stop, and manage Db2ZAI from Db2 Admin Tool.

Related information:

“Activating IBM Db2 AI for z/OS” on page 892
PH22698

Support for key labels

PH22698 - May, 2020

Db2 12 FL 502 Db2 12 function level 502 introduces key label management to support z/OS DFSMS transparent data set encryption. If you have function level 502 activated, you can use Db2 Admin Tool to manage key labels and perform operations on objects with key labels. For example, you can:

- Create a new table or storage group with a key label.
- Add or change a key label on an existing table or storage group.
- Generate DDL for objects with key labels.
- View key label information in the Db2 catalog.
- Compare objects with key labels.
- Mask key labels by using the new masks TBKEYLABL (for key labels on tables) and SGKEYLABL (for key labels on storage groups).

Related information:

"Creating tables" on page 334
"Adding or changing a key label" on page 469
"Altering a table by using the ALT line command" on page 461
"Mask definitions" on page 274
"Db2 catalog columns and the corresponding masks" on page 292
"Creating masks in a data set" on page 309
PH22698

New primary command to change the application compatibility level

PH22698 - May, 2020

You can use the new primary command APPLCOMPAT (or APPLC) to quickly change the application compatibility level. This command is equivalent to issuing the SET CURRENT APPLICATION COMPATIBILITY statement.

Related information:

"Db2 Admin Tool primary commands" on page 211
PH22698

Support for inserting partitions

PH20648 - April, 2020

Db2 12 (FL500) allows you to insert partitions between or before existing logical partitions in partition-by-range universal table spaces (UTS). Db2 Admin Tool supports this new functionality. You can insert partitions by using either the AL line command or ALT line command and then specifying the new INS line command. Additionally, when you generate SQL to re-recreate an object or compare objects, those operations can both generate SQL to insert partitions.

Related information:

"Altering a table by using the AL line command" on page 461
"Examples of redefining a table by using the ALT line command" on page 470
PH20648

CM batch option to generate JCL from a WSL

PH20257 - April, 2020

In Change Management (CM) batch, you can now generate JCL from a work statement list (WSL) by using the new parameter action_generate_jcl_from_wsl. The advantage of generating JCL is that you can review and edit the job before you submit it, whereas a WSL is not readable.

Related information:

"ACTION_GENERATE_JCL_FROM_WSL" on page 670
PH20257

Ability to view a change that supersedes a change

PH22548 - March, 2020

When using Change Management, you can view the change that supersedes a change by using the new SBY line command on panel ADB2C11.

If a change cannot be run because the status was switched to DEFINED by a change that supersedes it, a message is returned that identifies the superseding change.

Related information:

Db2 Administration Tool: Easier identification and control of object conflict in change management scenario (IBM community: Db2 Tools for z/OS)
“Viewing the change that supersedes a change” on page 845
“ADB9356E” on page 1176
PH22548

Exclusion of referential constraints when comparing objects with automatic target selection

PH22548 - March, 2020

This enhancement allows you to compare tables and apply the resulting changes without impacting referential integrity (RI) relationships. When you request a comparison with the following specifications, referential constraints are excluded from the comparison:

- specify the scope as table
- specify that the target is to be automatically selected
- request that the compare process run in CMDELTA mode
- do not suppress the drop of objects

In this case, referential constraints are not dropped in the target when the source does not include the referential constraints.

Related information:

PH22548

Conversion of ADB table spaces to UTS

PH22548 - March, 2020

Db2 12 FL 504 Beginning in Db2 12 function level 504, you can no longer create non-universal table spaces. This table space type is deprecated. To support this strategic direction to use only the universal table space (UTS) type, all Db2 Admin Tool (ADB) table spaces are converted to UTS when you run IBM Tools Customizer for z/OS (TCz) and select the option **Convert Admin table spaces to UTS**. The resulting customization job includes running the REORG utility to materialize the change for existing non-UTS table spaces.

Related information:

PH22548

Ability to view logical partition numbers and row counts for partitions

PH23202 - March, 2020

For partitioned tables and table spaces, you can now view logical partition numbers and the number of rows in each partition:

- On the **Table Space Parts for table-space (ADB21SP)** panel, you can now scroll to view all columns of the SYSTABLEPART catalog table for the selected table space, including the LOGICAL_PART and CARD (row counts) catalog columns.
- The table detail report is enhanced to include the logical partition numbers and row counts in the limit key section of the report. You can view this report by using the DET or BET commands on a partitioned table on the **Tables, Views, and Aliases (ADB21T)** panel.

Related information:

PH23202

Improvements to CM recovery

PH21635 - March, 2020

To ensure data integrity when using Change Management (CM), the process of generating and using recover changes is improved as follows:

- When generating a recover change during the analyze process, CM generates an UNLOAD statement for all tables that have column changes so that these tables can be recovered to the same point (with the existing or original data).
- When recovering with the original data, the data in the table is removed before the original data is loaded. This behavior prevents the duplication of data if ALTER operations were done as part of the recover change.
- When recovering multiple objects with the original data, all tables that need to be unloaded are placed in read-only mode at the same time so that the tables are logically consistent when they are recovered.

Related information:

"Analyzing a change" on page 652
"DATA_TO_RECOVER" on page 695
"RECOVER_OPTION" on page 710
PH21635

Bind Manager functionality added

PH18439 - March, 2020

The BNDAVB3 program from IBM Db2 Bind Manager for z/OS is integrated into Db2 Admin Tool and renamed ADBBMA3.

ADBBMA3 determines whether a bind is required for an application on which the bind avoidance program ADBBMAV has already been run and therefore, precompiling the application again is not necessary. (ADBBMAV runs the Db2 precompiler.) For example, if you run ADBBMAV on an application and then migrate that application to a different subsystem, only a bind is necessary. In this case, you can run ADBBMA3 on the application.

(ADBBMAV was previously integrated into Db2 Admin Tool by PH06267.)

Related information:

"Determining whether applications need to be rebound" on page 888
PH18439

Plan names displayed on ADB21K

PH16874 - March, 2020

The **Packages (ADB21K)** panel is enhanced to display the applicable plan name when displaying a package list for a plan.

Related information:

"Option K. Packages" on page 156
PH16874

ADBTEP2 can automatically retry failed statements

PH20650 - February, 2020

ADBTEP2 can automatically retry statements that fail due to timeouts when waiting on resources. To control retry behavior for ADBTEP2, use the new CM Batch parameters adbtep2_timeout_retries and adbtep2_timeout_wait_time or specify values on the **Batch Job Utility Parameters (ADB2UPA)** panel.

Related information:

Automatic ADBTEP2 retry for timeouts, deadlocks and unavailable resources (IBM Community: Db2 Tools for z/OS)
"ADBTEP2_TIMEOUT_RETRIES" on page 679
"ADBTEP2_TIMEOUT_WAIT_TIME" on page 679
"Batch job parameters for utility jobs" on page 605
"Parameters passed to the ADBTEP2 program" on page 573
PH20650

2019 new-function APARs for Db2 Admin Tool 12.1

The following APARs were delivered in 2019. They introduced enhancements to Db2 Admin Tool 12.1 since the General Availability (GA) date.

New warning when the compare scope is insufficient

PH19997 - December, 2019

When using IBM Db2 Object Comparison Tool for z/OS and the target of an object comparison is automatically selected, you can specify that a warning message be issued if the scope is insufficient. This message warns you that objects that exist only in the target might be dropped.

Related information:

"SCOPE_WARNING" on page 717
"ADB7353W" on page 1144
PH19997

Support for CONCENTRATESTMT and APREUSESOURCE bind options

PH19997 - December, 2019

Support is added for the new Db2 12 bind options CONCENTRATESTMT and APREUSESOURCE. These options can help improve query performance. CONCENTRATESTMT can be specified for the BIND PACKAGE and REBIND PACKAGE commands. APREUSESOURCE can be specified with the REBIND PACKAGE and REBIND TRIGGER PACKAGE commands.

Related information:

"Binding packages and generating BIND statements" on page 916
"Rebinding packages" on page 917
Improvements for the dynamic statement cache (Db2 12 for z/OS)
Static plan stability enhancements (Db2 12 for z/OS)
PH19997

Support for new FREE PACKAGE options

PH19997 - December, 2019

Support is added for the following new Db2 12 options for the FREE PACKAGE command:

- The INVALIDONLY option
- The ORIGINAL and PREVIOUS values for the PLANMGMTSCOPE option

Related information:

"Freeing packages" on page 919
PH19997

Ability to view package copy information

PH19997 - December, 2019

You can view information about previous and original copies of packages.

Related information:

“Viewing information about package copies” on page 924
PH19997

Support for INSERT ALGORITHM

PH16862 - October, 2019

Support is added for the new Db2 12 INSERT ALGORITHM attribute of a table space. You can perform the following actions in Db2 Admin Tool:

- You can specify this attribute when creating or altering a table space.
- You can also compare this field when doing object comparisons.
- You can mask or ignore this field.

Related information:

“Creating table spaces” on page 333
“Altering table spaces” on page 452
“Redefining partitions in a partitioned table space that uses index-controlled partitioning” on page 456
“Adding LOB columns to an existing table” on page 485
“Mask definitions” on page 274
“Db2 catalog columns and the corresponding masks” on page 292
“Verification masks” on page 289
“Ignores” on page 846
PH16862

Support for LOAD BACKOUT

PH16862 - October, 2019

Support is added for the new BACKOUT option of the LOAD utility with RESUME YES.

Related information:

“Running Db2 utilities on tables” on page 598
PH16862

Support for dynamic plan stability

PH16862 - October, 2019

Support is added for the new Db2 12 dynamic plan stability feature. You can perform the following actions in Db2 Admin Tool:

- Evaluate the cached dynamic statements that are likely to be captured and stabilized.
- Manage the captured and stabilized statements.
- Run EXPLAIN against a stabilized statement.
- Show objects that are associated with the stabilized statements.

Related information:

“Stabilizing dynamic SQL statements” on page 622
PH16862

Support for the UNLOAD privilege

PH10086 - October, 2019

Support is added for the new Db2 12 UNLOAD privilege is supported. You can grant and revoke this privilege. You can also see the impact of revoking the UNLOAD privilege for a table, including the impact on the privileges of other users (also known as cascade revokes). Additionally, if you call the GEN command for a table on which the UNLOAD privilege has been granted, the generated SQL includes the GRANT UNLOAD statement.

Related information:

"Granting authorizations" on page 257
"Revoking privileges" on page 351
PH10086

Support for REORG DROP_PART

PH10086 - October, 2019

Support is added for the new Db2 12 DROP_PART option for the REORG TABLESPACE utility. In Db2 Admin Tool, you can specify this new utility option either as a CM batch parameter (util_reorg_drop_part) or on **Specify Utility Options - REORG (ADB2USO)** panel.

Related information:

"UTIL_REORG_DROP_PART" on page 764
PH10086

CM uses DSNUTILV instead of DSNUTILU

PH10086 - October, 2019

If you are using Db2 12, the Change Management (CM) batch interface now uses the new DSNUTILV stored procedure to run utilities. Therefore, you can specify utility statements that are larger than 32 KB. (If you are using a prior version of Db2 for z/OS, CM batch uses the DSNUTILU stored procedure to run utilities.)

Related information:

"Managing Changes by using the CM batch interface" on page 659
PH10086

Ability to specify how CM runs utilities

PH10086 - October, 2019

When using the Change Management (CM) batch interface, you can specify how to run Db2 utilities. They can be run either by calling a Db2 stored procedure (DSNUTILV for Db2 12 or DSNUTILU for earlier versions of Db2) or by invoking them directly in the CM batch job. Use the new CM batch parameter use_dsnutil_sp to control this behavior.

Related information:

CM Batch September 2019 enhancements: reduce contention and run utilities inline (IBM community: Db2 Tools for z/OS)
use_dsnutil_sp
PH10086

Support for DELETE FETCH FIRST *n* ROWS ONLY

PH10086 - October, 2019

Support is added for the new Db2 12 FETCH FIRST *n* ROWS ONLY clause in an SQL DELETE statement. This clause limits the number of rows that are deleted in a single SQL statement. In Db2 Admin Tool, you can add this clause to a DELETE statement by using the new FETCH command on the **Build SQL DELETE Prototype (ADB21TDE)** panel.

Related information:

"Build SQL Prototype panel" on page 327
PH10086

Support for new SELECT clauses

PH10086 - October, 2019

Support is added for the following new Db2 12 clauses in SELECT statements: OFFSET, FETCH NEXT (or FETCH FIRST) and LIMIT

Related information:

PH10086

Reduced resource contention in CM

PH10086 - October, 2019

Change Management (CM) has been enhanced to reduce resource contention and the likelihood of a timeout when running a change. Specifically, a commit was added to release locks after updating a change. Also, the isolation level of the change table ADBSCHG was changed to ROW to reduce contention between two changes.

Related information:

CM Batch September 2019 enhancements: reduce contention and run utilities inline (IBM community: Db2 Tools for z/OS)
PH10086

Support for the COMPRESSRATIO column in SYSTABLESPACE

PH10086 - October, 2019

The new Db2 12 catalog column COMPRESSRATIO is included in the relevant table space panels:

- **Table Spaces (ADB21S)** panel
- **Interpretation of an Object in SYSTABLESPACE (ADB21SI1)** panel

Related information:

PH10086

APPLCOMPAT V12R1M505

PH10086 - October, 2019

Db2 12 FL 505 The APPLCOMPAT bind option V12R1M505 is supported.

Related information:

PH10086

Support for updated IDAA trace details

PH09487 - July, 2019

If you are using IBM Db2 Analytics Accelerator for z/OS 7.1.0 (IDAA 7.1) or later, you can view the updated trace details in Db2 Admin Tool.

Related information:

"Retrieving trace data for accelerators" on page 944
"Displaying accelerator status" on page 935

PH09487

Support for the INVALIDATECACHE and USE PROFILE statistics options

PH02457 - July, 2019

You can specify the following utility options on the **Specify Utility Options** panel when gathering inline statistics:

- INVALIDATECACHE (for the REORG, LOAD, and REBUILD INDEX utilities)
- USE PROFILE (for the LOAD and REORG TABLSPACE utilities)

When using the Change Management (CM) batch interface, you can specify the following utility statistics options:

- INVALIDATECACHE (for REORG and RUNSTATS)
- USE PROFILE (for REORG)

Related information:

"Running Db2 utilities on tables" on page 598
util_reorg_statistics_invalidatecache
util_reorg_statistics_use_profile
util_runstats_invalidatecache
PH02457

Db2 Admin Tool dynamically finds the DECP settings

PH02457 - July, 2019

Db2 Admin Tool now dynamically finds DECP settings by using Db2 session variables instead of getting them from DSNHDECP module. Therefore, Db2 Admin Tool can get DECP settings regardless of the name of the DSNHDECP module.

Related information:

PH02457

Improved function level messages

PH12055 - June, 2019

Messages about Db2 12 function levels are improved.

Related information:

"Support for Db2 continuous delivery" on page 69
"ADB1970W" on page 1096
"ADB1971S" on page 1097
"ADB1972W" on page 1097
PH12055

Improvements to ADBMSGS

PH12055 - June, 2019

Object comparison reports are included in the ADBMSGS data set.

Related information:

"Consolidating messages into a single file" on page 246
"ADB8998I" on page 1165
"ADB8999I" on page 1166

PH12055

Support for Db2 12 continuous delivery

PH06164 - May, 2019

Support is added for Db2 12 continuous delivery.

Related information:

“Support for Db2 continuous delivery” on page 69
PH06164

APPLCOMPAT values for function levels

PH06164 - May, 2019

For the APPLCOMPAT bind option, you can specify values for Db2 12 function levels. (At the time of this enhancement, valid values are up to V12R1M504.)

Related information:

“Binding packages and generating BIND statements” on page 916
“Rebinding packages” on page 917
PH06164

Invocation exit to set global variables

PH09606 - May, 2019

You can set global variables by using an invocation exit.

Related information:

Implementing site standards in IBM Db2 Administration Tool (IBM community: Db2 Tools for z/OS)
IBM Db2 Administration Tool: Site standards follow-up (IBM community: Db2 Tools for z/OS)
“Setting global variables for Db2 Admin Tool” on page 125
PH09606

Readability improvements to ADBMSGS

PH08484 - May, 2019

The ADBMSGS data set contains the consolidated messages from batch jobs. The format of the messages in this data set is improved for readability.

Related information:

“Consolidating messages into a single file” on page 246
Admin & OC: Consolidated messages from CM Batch and beyond (IBM community: Db2 Tools for z/OS)
Db2 Object Comparison Tool: ADBMSGS filtering (IBM community: Db2 Tools for z/OS)
PH08484

CM can unload tables when analyzing changes

PH00552 - April, 2019

When using Change Management (CM), you can specify whether to unload altered tables as part of the analyze process. Specify your preference by using the new CM batch parameter unload_altered_tables or the **Unload altered tables** field on the **Options for Change Functions (ADB2PCO)** panel

Related information:

“UNLOAD_ALTERED_TABLES” on page 727

“Enabling and disabling automatic recreate, reload, or removal of accelerated tables” on page 953
(See panel ADB2PCO)
PH00552

CM support for partition-level copies

PH03675 - February, 2019

You can use the Change Manage (CM) batch interface to generate COPY utility statements for partition-level image copies.

Related information:

util_listdef_partlevel
PH03675

2018 new-function APARs for Db2 Admin Tool 12.1

The following APARs were delivered in 2018. They introduced enhancements to Db2 Admin Tool 12.1 since the General Availability (GA) date.

Bind avoidance

PH06267 - December, 2018

The bind avoidance feature of Db2 Bind Manager is integrated into Db2 Admin Tool.

Related information:

“Determining whether applications need to be rebound” on page 888
PH06267

DBRM regeneration

PH06267 - December, 2018

The DBRM regeneration feature of Db2 Bind Manager is integrated into Db2 Admin Tool.

Related information:

“ Regenerating DBRMs” on page 927
PH06267

Collection clean-up

PH06267 - December, 2018

The collection clean-up feature of Db2 Bind Manager is integrated into Db2 Admin Tool.

Related information:

“Deleting obsolete packages” on page 925
PH06267

Ability to view object details in batch mode

PI96053 - May, 2018

You can retrieve details about objects in batch mode.

Related information:

“Details about objects in batch mode” on page 507
PI96053

Ability to validate imported DDL in CM batch

PI96053 - May, 2018

You can validate DDL that is imported during CM batch.

Related information:

“VALIDATE_DDL” on page 743
Improved validation of input DDL for CM Batch (IBM community: Db2 Tools for z/OS)
PI96053

CM batch parameters for readable WSL

PI96053 - May, 2018

You can use new CM batch parameters to specify a data set where the readable version of a work statement list (WSL) is stored.

Related information:

“ACTION_CONVERT_TO_ISPF_WSL” on page 667
“PDS_FOR_WSL_CONV ” on page 709
“WORKLIST_NAME_CONV” on page 744
Db2 Administration Tool: Convert work statement lists to readable format with CM Batch (IBM community: Db2 Tools for z/OS)
PI96053

Ability to alter sequence aliases

PI92080 - March, 2018

You can alter sequence aliases.

Related information:

“Altering sequence aliases” on page 499
PI92080

Ability to define restart points in a WSL

PI61852 - March, 2018

When you run a work statement lists (WSL) by using ADBTEP2 in CM batch, you can restart a change at a specific point in the WSL that you define instead of only at the point where it failed. This functionality gives you more control over where the WSL is restarted.

Related information:

“ADBTEP2_RESTART” on page 674 (the U value)
PI61852

Consolidation of messages into one file

PI66475 - March, 2018

You can specify that all messages are consolidated into a single output file so that they are easier to find. You can define this output file by using the new ADBMSGS DD statement.

Related information:

“Consolidating messages into a single file” on page 246
PI66475

Improvements to CM batch

PI74777 - March, 2018

By using the CM batch interface, you can run compare and generate and run a WSL without going through the change management processes.

Related information:

“ACTION_GENERATE_WSL” on page 670
“ACTION_RUN_WSL” on page 672
PI74777

UTILFROM ADDPART

PI80511- March, 2018

When using ADBTEP2 to execute the LOAD utility, you can add discard data set information at the partition level. To do, use the new ADDPART parameter in the UTILFROM utility.

Related information:

“UTILFROM statements” on page 532
PI80511

New messages for CCSID changes

PI82330 - March, 2018

When the ALT/TS CCSID attribute is changed and associated tables are involved in an RI relationship, message ADBU004W is issued.

When the CCSID attribute of a column that is part of a key in a referential constraint is changed, message ADBP015W is issued.

Related information:

“ADBU004W” on page 1194
“ADBP015W” on page 1193
PI82330

Db2 12 function level support

When you activate new Db2 12 function levels in a Db2 subsystem or data sharing group, enhancements might become available that impact Db2 Admin Tool.

The following function levels are tolerated or supported by Db2 Admin Tool and are provided with the corresponding PTF, if any.

Table 2. Db2 Admin Tool PTFs in support of Db2 12 function levels

Db2 12 function level	Toleration PTF	Support PTF
FL 510	UI75519 (APAR PH36485)	UI75519 (APAR PH36485)

Table 2. Db2 Admin Tool PTFs in support of Db2 12 function levels (continued)

Db2 12 function level	Toleration PTF	Support PTF
FL 509	UI73968 (APAR PH34152)	UI75241 (APAR PH36481) UI75518 (APAR PH36482) UI75519 (APAR PH36485) "Support for specifying a compression algorithm at the object level" on page 37 "Support for high availability for accelerator-only tables" on page 37 "Support for tamper-proof audit policies" on page 37 Video: Db2 Administration Tool: FL 509 support
FL 508	UI72139 (APAR PH30133)	UI73496 (APAR PH31554) UI76030 (APAR PH36790) Support for moving tables from multi-table table spaces to UTS Usability enhancements for function level 508 Video: Db2 Administration Tool: moving tables from multi-table table spaces into universal table spaces (FL508) (IBM community: Db2 Tools for z/OS)
FL 507	UI70406 (APAR PH25983)	UI71802 (APAR PH28647) UI72418 (APAR PH29663) Support for Db2 12 function level 507
FL 506	UI66622 (APAR PH18697)	UI70405 (APAR PH25686) UI72032 (APAR PH27088) Support for Db2 12 function level 506
FL 505	UI64651 (APAR PH02457)	UI69881 (APAR PH24230) Support for Db2 12 function level 505

Table 2. Db2 Admin Tool PTFs in support of Db2 12 function levels (continued)

Db2 12 function level	Toleration PTF	Support PTF
FL 504	UI64042 (APAR PH12055)	UI69880 (APAR PH22951) UI70958 (APAR PH27083) UI72419 (APAR PH28566) Support for objects that you can no longer create in Db2 12 function level 504 Video: Db2 Administration Tool: FL 504 support
FL 503	UI64042 (APAR PH12055)	UI69677 (APAR PH22698) Support for IBM Db2 AI for z/OS
FL 502	UI64042 (APAR PH12055)	UI69677 (APAR PH22698) Support for key labels
FL 501	UI62635 (APAR PH06164)	UI69451(APAR PH24975)
FL 500	UI62635 (APAR PH06164)	UI69451(APAR PH24975)

The following function levels are tolerated or supported by Object Comparison Tool (an extension to Db2 Admin Tool) and are provided with the corresponding PTF, if any.

Table 3. Object Comparison Tool PTFs in support of Db2 12 function levels

Db2 12 function level	Toleration PTF [1]	Support PTF [1]
FL 510	No additional PTF required	No additional PTF required
FL 509	No additional PTF required	No additional PTF required
FL 508	No additional PTF required	No additional PTF required
FL 507	No additional PTF required	No additional PTF required
FL 506	No additional PTF required	No additional PTF required
FL 505	UI64652 (APAR PH14718)	No additional PTF required
FL 504	No additional PTF required	UI69879 (APAR PH25685)
FL 503	No additional PTF required	No additional PTF required
FL 502	No additional PTF required	No additional PTF required
FL 501	No additional PTF required	UI69452 (APAR PH24976)
FL 500	No additional PTF required	UI69452 (APAR PH24976)

Note:

1. Because Object Comparison Tool runs as an extension of Db2 Admin Tool, any PTF that is required by Db2 Admin Tool for a particular function level is also required by Object Comparison Tool.

Related information

"Function level support terminology" on page 69

Changed messages

Occasionally, message text and suffix values might change. If you have code that checks for message text or numbers, use the list of changed messages to determine if you need to make any updates.

The following Db2 Admin Tool messages have changed meanings or other significant technical changes since the GA of version 12.1:

ADB3001E

With APAR PH58340, this message now includes the following text if needed:

```
Object was dropped and not recreated
```

or

```
Object was implicitly dropped and not recreated
```

ADB3002E

With APAR PH58340, this message now includes the following text if needed:

```
Object was dropped and not recreated
```

or

```
Object was implicitly dropped and not recreated
```

ADB7198W

Previously this message was issued when altered tables were unloaded during the analyze process. With APAR PH36418, this message is now issued when altered tables are *not* unloaded during the analyze process. Additionally, this message has changed from an informational message (ADB7198I) to a warning message (ADB7198W).

ADB7385W

With APAR PH36418, this message has changed from an informational message (ADB7385I) to a warning message (ADB7385W). Additionally, the text of this message has changed as follows:

Old message text:

```
Option "Recreate for NULLS Change" is only applicable when "Unload Altered tables" is set to
YES.
"Recreate for NULLS Change" will be given the default value, YES.
```

New message text:

```
The Preserve all data (preserve_all_data) option was not activated for this compare process.
Changes to whether a column can contain null values will be processed by using ALTER
statements, which
may result in data loss.
```

ADB7389W

With APAR PH34842, this message has changed from an informational message (ADB7389I) to a warning message (ADB7389W).

Deprecated functions and functions that are no longer supported in Db2 Admin Tool 12.1

Plan to remove any dependencies on functions that are deprecated or no longer supported in Db2 Admin Tool 12.1.

Deprecated function

Certain capabilities of IBM Db2 Administration Tool for z/OS 12.1 are *deprecated*, meaning that their use is discouraged. Support is likely to be removed in the future. Avoid creating new dependencies that rely on any deprecated function(s), and develop plans to remove any dependencies on such function(s).

Deprecated function	Recommended alternative	Deprecated date	Target removal date

System-level backups:
System-level backups are created by the Db2 BACKUP SYSTEM utility and recovered by the Db2 RESTORE SYSTEM utility. Currently, you can manage system-level backups in Db2 Admin Tool by using the following options on the **System Administration (ADB2Z)** panel:
`System Backup and Recovery:`
`SB - Backup System`
`SR - Recover System`
`PT - Set Point in Time`
These options are deprecated and might be removed in the future.
	Use the system-level backup capabilities in IBM Db2 Recovery Expert for z/OS. See IBM Db2 Recovery Expert for z/OS.	March 27, 2020	To be determined

Authorization switching: Authorization switching is a facility within Db2 Admin Tool that is used to execute DDL and DCL under the authority of another user. This entire facility is deprecated.	Use RUN SQLID or trusted contexts as follows: • If you want objects to be owned or managed by a secondary authorization ID (or RACF group) with SYSADM authority, use Run SQLID. When you specify a value for Run SQLID, Db2 Admin Tool adds a SET CURRENT SQLID=*runsqlid* statement to the beginning of the change and runs everything with that SQLID. This scenario requires that the Db2 security exit define the SQLID as a secondary authorization ID for the user that submits the job. For a RACF group, the user must be connected to that group. • You can use a trusted context in Db2 Admin Tool if you want changes to be made only by using a trusted context. In this case, the DBAs do not have SYSADM privilege by using a privilege on a secondary authorization ID. For more information, see “Option TR. Trusted Contexts” on page 183.	March 26, 2021	To be determined

Function that Db2 Admin Tool 12.1 no longer supports

In IBM Tools Customizer for z/OS (TCz) for Db2 Admin Tool, the InfoSphere Optim™ Configuration Manager (OCM) option has been removed, because OCM has reached its End of Support (EOS) date. For details, see the OCM lifecycle information: https://www.ibm.com/support/pages/lifecycle/search?q=IBM%20InfoSphere%20Optim%20Configuration%20Manager

Db2 Admin Tool support for migrating to Db2 12

When you migrate to Db2 12 for z/OS, you need to take some specific actions in Db2 Admin Tool to support this new release. Additionally, you can use Db2 Admin Tool to perform certain Db2 12 migration actions, such as converting deprecated objects and finding old packages, and to help accelerate your timeline for using new Db2 12 features.

Use the following information to help you when you migrate to Db2 12:

- **"Checklist: Actions to take in Db2 Admin Tool when you migrate to Db2 12" on page 66**
- **"Migration questions and answers:" on page 67**
  - "For Db2 12 FL 100, what maintenance is required for Db2 Admin Tool?" on page 67
  - "What is the difference between a tolerated function level and a supported function level?" on page 67
  - "What APPLCOMPAT value should I specify for Db2 Admin Tool packages?" on page 67
  - "Do I need to run TCz for Db2 Admin Tool when I activate a new Db2 12 function level?" on page 67
- **"Ways that Db2 Admin Tool can help you migrate to Db2 12:" on page 67**
  - "Converting non-universal table spaces to the universal table space (UTS) type by using Db2 Admin Tool" on page 67
  - "Using Db2 Admin Tool to identify Db2 packages that were bound prior to Db2 10" on page 67
  - "Identifying whether your system is approaching the 6-byte RBA limit" on page 68
- **"Db2 Admin Tool support for Db2 12 enhancements" on page 69**
- **"Additional Support" on page 69**

Checklist: Actions to take in Db2 Admin Tool when you migrate to Db2 12

1. **Upgrade to Db2 Admin Tool 12.1.**

   Db2 Admin Tool 11.2 does not support Db2 12. Additionally, Db2 Admin Tool 11.2 has passed its End of Support (EOS) date. See Product Lifecycle for DB2 Tools (IBM Support)

2. **Apply fixes for Db2 12 support.**

   Apply the following fixes to Db2 Admin Tool to ensure support for the new Db2 12 continuous delivery framework and for several new objects:

Db2 Admin Tool APAR	Description
PH06164	"Support for Db2 12 continuous delivery" on page 58 "APPLCOMPAT values for function levels" on page 58
PH12055	"Improved function level messages" on page 57
PH10086	"CM uses DSNUTILV instead of DSNUTILU" on page 55
PH22548	"Conversion of ADB table spaces to UTS" on page 51

3. **Apply fixes for function level support.**

   Apply the appropriate fixes to ensure that Db2 Admin Tool supports the Db2 12 function level that you plan to use. For a list those fixes, see "Db2 12 function level support" on page 61.

4. **Run IBM Tools Customizer for z/OS (TCz) for Db2 Admin Tool.**

   You must run TCz for Db2 Admin Tool to specify the migrated Db2 12 libraries. To do so, complete the steps in "Updating Db2 Admin Tool after migrating to a new Db2 version, mode, or function level" on page 98. These steps guide you through the process of regenerating and running the jobs that are required to rebind the necessary ADB plans and packages. These steps also include the specific TCz settings for Db2 12.

Migration questions and answers:

For Db2 12 FL 100, what maintenance is required for Db2 Admin Tool?

Db2 Admin Tool does not require a minimum PTF level for FL 100. However, in preparation for moving to FL 500, IBM recommends that you apply PTF UI62635 (APAR PH06164).

What is the difference between a tolerated function level and a supported function level?

A *tolerated function level* means that you can run Db2 Admin Tool, but not all enhancements in that function level are guaranteed to be supported. A *supported function level* means that you can use the enhancements in that function level with Db2 Admin Tool.

For more information about the maintenance that is required for Db2 Admin Tool to support each function level, see "Db2 12 function level support" on page 61.

What APPLCOMPAT value should I specify for Db2 Admin Tool packages?

When you rebind Db2 Admin Tool packages, the APPLCOMPAT value that you specify must be less than or equal to the currently activated Db2 function level for your Db2 subsystem or data sharing group:

- The safest value to specify for the APPLCOMPAT bind option is the highest function level that is currently supported by Db2 Admin Tool. This value prevents users from exploiting features until they are fully supported by Db2 Admin Tool.
- If you rebind at a function level that is higher than the highest supported function level, you can manually run your DDL through ISPF panels or execute a work statement list (WSL) that exploits features at that function level.
- If you rebind at a function level that is lower than that highest supported function level, you must set the target function level (tgt_db2fl) to the same (or lower) function level that you specified for the Db2 Admin Tool packages. Otherwise, Db2 Admin Tool might generate DDL that subsequently cannot be run.
- When the version of Db2 Admin Tool is 12.1 and the version of Db2 is 12, the lowest APPCOMPAT value that you can specify is V11R1.

Do I need to run TCz for Db2 Admin Tool when I activate a new Db2 12 function level?

Yes. Complete the steps that are described in "Updating Db2 Admin Tool after migrating to a new Db2 version, mode, or function level" on page 98. You must take these actions in TCz to account for any Db2 catalog changes and to be able to use the enhancements in that function level.

Ways that Db2 Admin Tool can help you migrate to Db2 12:

Converting non-universal table spaces to the universal table space (UTS) type by using Db2 Admin Tool

Converting to UTS can help you take advantage of the growing list of Db2 12 features that operate only on UTS. You can use Db2 Admin Tool to help you convert partitioned table spaces to PBR UTS. See Convert a group of partitioned table spaces to PBR UTS.

You can also use Db2 Admin Tool to move tables from deprecated multi-table table spaces to UTS. See "Moving tables from multi-table table spaces to UTS" on page 503.

Using Db2 Admin Tool to identify Db2 packages that were bound prior to Db2 10

When you migrate to Db2 12, any packages that were bound prior to Db2 10 can cause a problem. If execution of these packages is requested in a Db2 12 environment, Db2 automatically rebinds them. Because these autobinds can be disruptive to your application workload, you should identify and rebind these packages prior to migrating to Db2 12. One way to find these packages is to customize and run the Db2 job DSNTIJPM. However, an easier alternative is to use theDb2 Admin Tool interface.

To identify packages that were bound before Db2 10:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.

2. At the bottom of the **System Catalog (ADB21)** panel, specify the following filtering criteria:

```
Enter standard selection criteria: Settings: LIKE operator; Criteria saved.
 Name  . . . .                            > Grantor . . .          >
 Schema  . . .                            > Grantee . . .          >
 Owner . . . .                            >
 In DB/Coll  .                            > Switch Catalog Copy   . . N (N/S/C)
 And/or other selection criteria (option xC shows you columns for option x)
 Column  . . . RELBOUND            > Operator  . . <      Value  . . O
```

3. Specify option K, and press Enter.

   The **Packages (ADB21K)** panel displays all packages that were bound prior to Db2 10 and potentially need to be rebound.

4. To determine when a package was last used, scroll to the right and look at the value in the **Last Used** column. This information can help you determine if a package is likely be executed and thus automatically rebound in your Db2 12 environment.

5. For more details on a particular package, use the DET line command to display the **Details for object(s) (ADBPD)** panel:

```
ADBPD min ----------------- DC1A Details for object(s) ---------------- 18:43

Commands: SAVE  ZOOM

  Details for package : ADB_MT_CHANGE_UPDATE      in collection : TS6462

   Package information
    Package type . . . . . . . . . . : Trigger package
    Version  . . . . . . . . . . . . :
    Authorization ID of owner  . . . : TS6462
    Owner type . . . . . . . . . . . : Auth ID
    Authorization ID of creator  . . : TS6462
    Created timestamp  . . . . . . . : 2020-05-29-18.01.17.917089
    Latest BIND timestamp  . . . . . : 2020-05-29-18.01.17.917089
    Version under which package bound: V12
    Qualifier for unqualified SQL  . : TS6462
    Operative status of package  . . : Operative, but REBIND will take place
    Resource and authorization check : At BIND time
    Size of the base section (bytes) : 3824   (in EDM pool during execution)
    Average DML section size (bytes) : 12512  (loaded when needed during exec)
    Package bound with EXPLAIN . . . : No
    ...
```

Identifying whether your system is approaching the 6-byte RBA limit

When you migrate to Db2 12, all database objects eventually need to be converted from using 6-byte RBA or LRSN values (basic format) to accommodate 10-byte RBA values (extended format). Typically, this conversion is done by running the REORG utility. If you are not running in data sharing mode and your system's high RBA is getting close to the limit for 6-byte representation, you need to convert your database objects soon. If the RBA value passes the 6-byte threshold, objects in basic format cannot be updated until they are converted to the extended format.

You can use Db2 Admin Tool to help you determine if your system is getting close to the 6-byte RBA limit by taking the following actions:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option `LI`, and press Enter to issue the Db2 DIS LOG command.
3. In the output, look at the RBA of the last written log record to determine if you are nearing the 6-byte limit:

```
DB2 Admin ------- DC1Q Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                                 Scroll ===> PAGE

 -DIS LOG


********************************* Top of Data **********************************
DSNJ370I  ¤DC1Q DSNJC00A LOG DISPLAY
CURRENT COPY1 LOG = DC1Q.LOGCOPY1.DS02 IS 41% FULL
CURRENT COPY2 LOG = DC1Q.LOGCOPY2.DS02 IS 41% FULL
          H/W RBA = 0000000010ACD1BBA9B3
          H/O RBA = 0000000010ACD0DF6FFF
          FULL LOGS TO OFFLOAD = 0 OF 6
          OFFLOAD TASK IS (AVAILABLE)
          SOFTWARE ACCELERATION IS DISABLED
          ZHYPERLINK WRITE IS DISABLED
DSNJ371I  ¤DC1Q DB2 RESTARTED 12:08:20 JUN  6, 2020
          RESTART RBA 0000000010A413B04000
          CHECKPOINT FREQUENCY 1000000 LOGRECORDS
          LAST SYSTEM CHECKPOINT TAKEN 07:24:15 JUN 16, 2020
DSN9022I  ¤DC1Q DSNJC001 '-DIS LOG' NORMAL COMPLETION
******************************** Bottom of Data ********************************
```

For more information about RBA and LRSN limits, see What to do before RBA or LRSN limits are reached (Db2 12 for z/OS).

Db2 Admin Tool support for Db2 12 enhancements

When you develop a Db2 migration plan, you likely set target dates for using the new functionality that is introduced in the release. Db2 Admin Tool can potentially help you move that date forward. Because Db2 Admin Tool supports Db2 12 enhancements, it can help you quickly adopt new features such as insert algorithm 2 for improved performance and the ability to insert partitions between existing logical partitions.

Db2 Admin Tool supports features that are available in the base release and features that are available in function levels. For the latest information about the features that Db2 Admin Tool supports in each function level, see “Db2 12 function level support” on page 61.

Additional Support

If you have additional questions or need more help when migrating to Db2 12, please contact IBM Software Support. IBM is ready to help you successfully migrate to Db2 12 and use Db2 Admin Tool to more easily manage your data.

Support for Db2 continuous delivery

Db2 12 for z/OS introduced the concept of a function level to support continuous delivery of new enhancements. A *function level* is a single PTF that enables the activation of a specific set of Db2 enhancements. Db2 Admin Tool support for each function level is delivered in the maintenance stream.

If all current maintenance is applied, Db2 Admin Tool 12.1 tolerates all Db2 12 function levels and supports them as soon as possible. If you are not current on maintenance for Db2 Admin Tool, certain function levels might not be tolerated or supported.

For a list of toleration and support PTFs for each function level, see “Db2 12 function level support” on page 61.

Recommendation: Stay current on maintenance to ensure the latest function level support.

Function level support terminology

Levels of support for Db2 12 function levels are defined as follows:

Supported function level
You can use the enhancements in this function level with Db2 Admin Tool.

Tolerated function level

Db2 Admin Tool can run on a member or subsystem at this function level, but not all enhancements in the function level are guaranteed to be supported.

If you run Db2 Admin Tool on a subsystem with a function level that is tolerated but not supported and use enhancements in that function level, affected objects or attributes might not be handled or displayed correctly. For example, if an object is dropped and recreated to implement a change, new attributes for that object might not be defined. If you do not use new enhancements in the tolerated function level, Db2 Admin Tool runs normally.

Function level that is not tolerated

If you try to run Db2 Admin Tool on a member or subsystem at this function level, results might be unpredictable. However, most Db2 Admin Tool features work.

Db2 function level confirmation when starting Db2 Admin Tool

When Db2 Admin Tool starts, if the current function level is not supported or tolerated, a confirmation panel is displayed, as shown in the following example:

```
ADB2CONF          DC1A Db2 Function Level Confirmation              15:19

The Db2 Administration Tool tolerates function level 506.

However, enhancements in this function level are not supported.
If you do not use any of these enhancements, Db2 Admin operates normally.

Select a choice
  1.  Exit
  2.  Continue - I understand and accept the risk for FL enhancements
  3.  Continue as above and do not prompt again for this function level.
```

In this example, function level 506 is activated. The Db2 Admin Tool toleration PTF for FL 506 has been applied, but the support PTF for FL 506 has not been applied. If you continue, Db2 Admin Tool works. However, you cannot use enhancements in function level 506.

If you want to continue, select option **2** or **3**. If you select either of these options, you accept that you cannot use the enhancements in that function level. If you specify option **2**, your choice is set for only the current session. If you select option **3**, your choice is stored in the ISPF profile. When you specify one of these choices, Db2 Admin Tool sets the "Maximum Db2 function level accepted" on page 73 to the current function level.

Related information

Adopting new capabilities in Db2 12 continuous delivery (Db2 12 for z/OS)

Db2 12 function levels (Db2 12 for z/OS)

Db2 function level settings in Db2 Admin Tool

Within Db2 Admin Tool, you can specify the Db2 function level, the APPLCOMPAT function level, the target function level, and the maximum Db2 function level accepted.

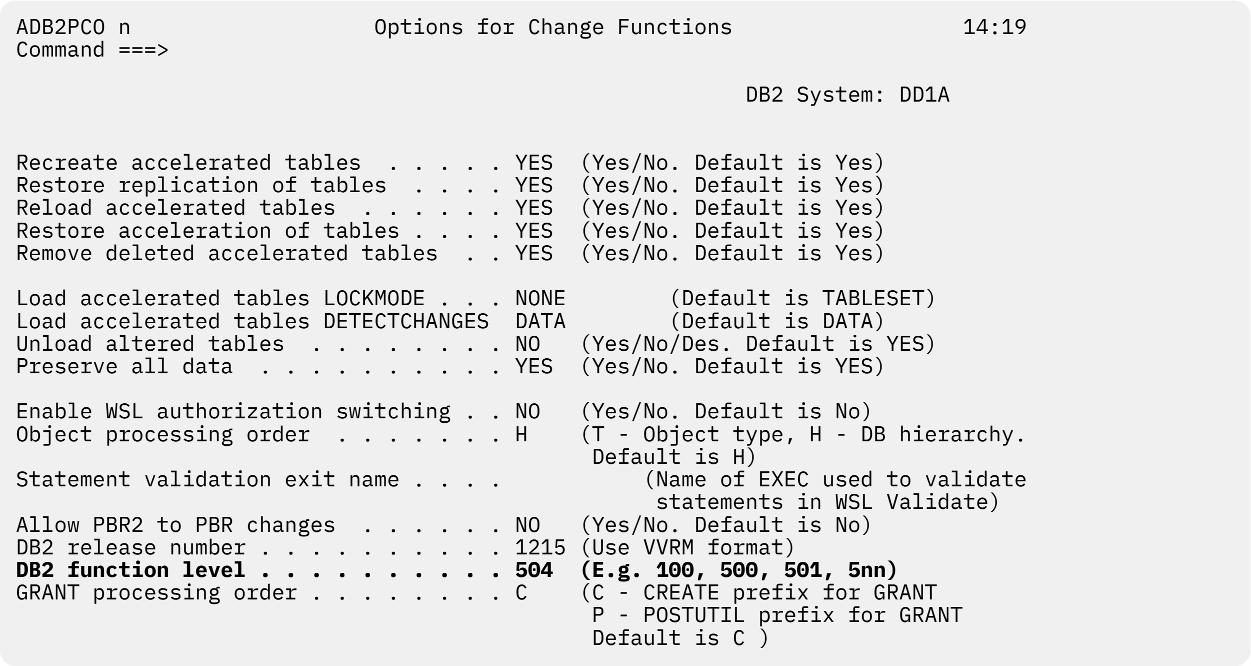
These terms are defined as follows:

Db2 function level

The function level at which the Db2 subsystem is currently running.

When Db2 Admin Tool starts, it gets this value from Db2. For batch jobs that do not connect to Db2, Db2 Admin Tool determines the current function level of the subsystem based on the value in the DSNHDECP module.

You can override this value by specifying a different function level on the **Options for Change Functions (ADB2PCO)** panel in the **DB2 function level** field:



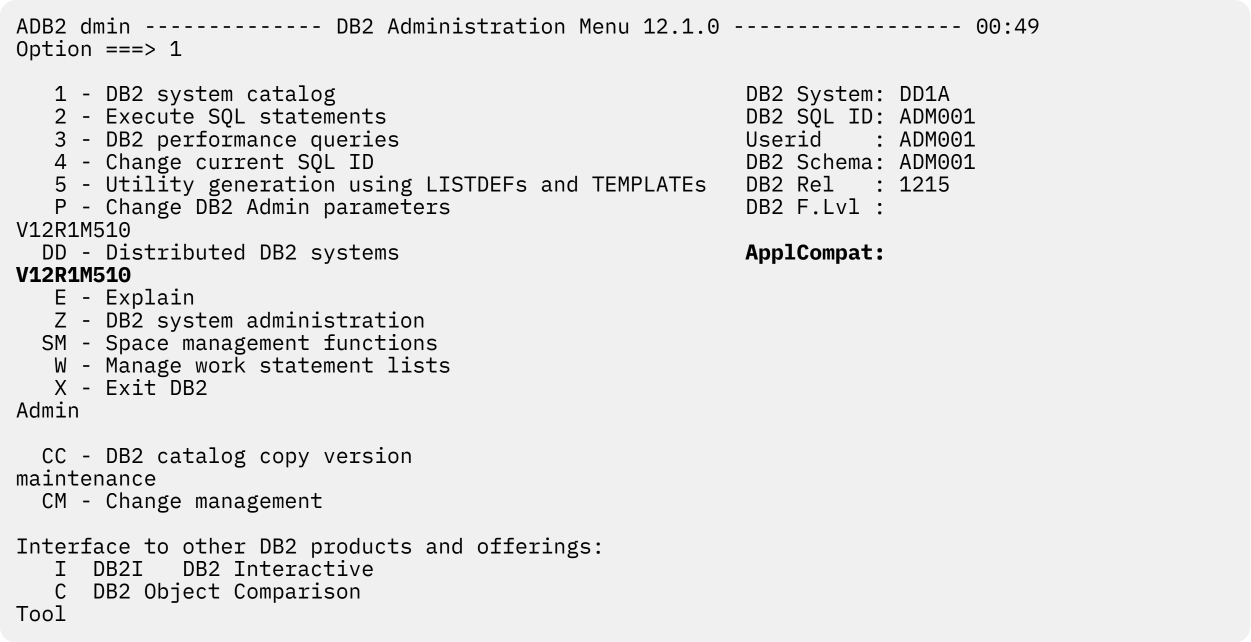
```
ADB2PCO n                   Options for Change Functions                  14:19
Command ===>

                                                         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 LOCKMODE . . . NONE        (Default is TABLESET)
Load accelerated tables DETECTCHANGES  DATA        (Default is DATA)
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 . . . . . . . . . . 1215 (Use VVRM format)
DB2 function level . . . . . . . . . . 504  (E.g. 100, 500, 501, 5nn)
GRANT processing order . . . . . . . . C    (C - CREATE prefix for GRANT
                                             P - POSTUTIL prefix for GRANT
                                             Default is C )
```

*Figure 6. Db2 function level on the **Options for Change Functions (ADB2PCO)** panel*

Tip: To navigate to this panel, specify `P.CH` on the **DB2 Administration Menu (ADB2)** panel.

APPLCOMPAT function level

The function level value of the CURRENT APPLICATION COMPATIBILITY special register. This value is displayed on the main **DB2 Administration Menu (ADB2)** panel:



```
ADB2 dmin -------------- DB2 Administration Menu 12.1.0 ------------------ 00:49
Option ===> 1

   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   : 1215
   P - Change DB2 Admin parameters                       DB2 F.Lvl :
V12R1M510
  DD - Distributed DB2 systems                           ApplCompat:
V12R1M510
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2
Admin

  CC - DB2 catalog copy version
maintenance
  CM - Change management

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

*Figure 7. APPLCOMPAT value on the **DB2 Administration Menu (ADB2)** panel*

The value that is displayed on this panel is the function level that is used by Db2 Admin Tool for all functions, except for generating SQL. For generating SQL, Db2 Admin Tool uses the target function level.

The APPLCOMPAT function level is initially set to the APPLCOMPAT value with which the ADBMAIN package was bound. If you did not specify an APPLCOMPAT bind value, this value is set to the “Db2 function level” on page 70.

You can update the APPLCOMPAT function level by using a SET statement or the APPLCOMPAT primary command.

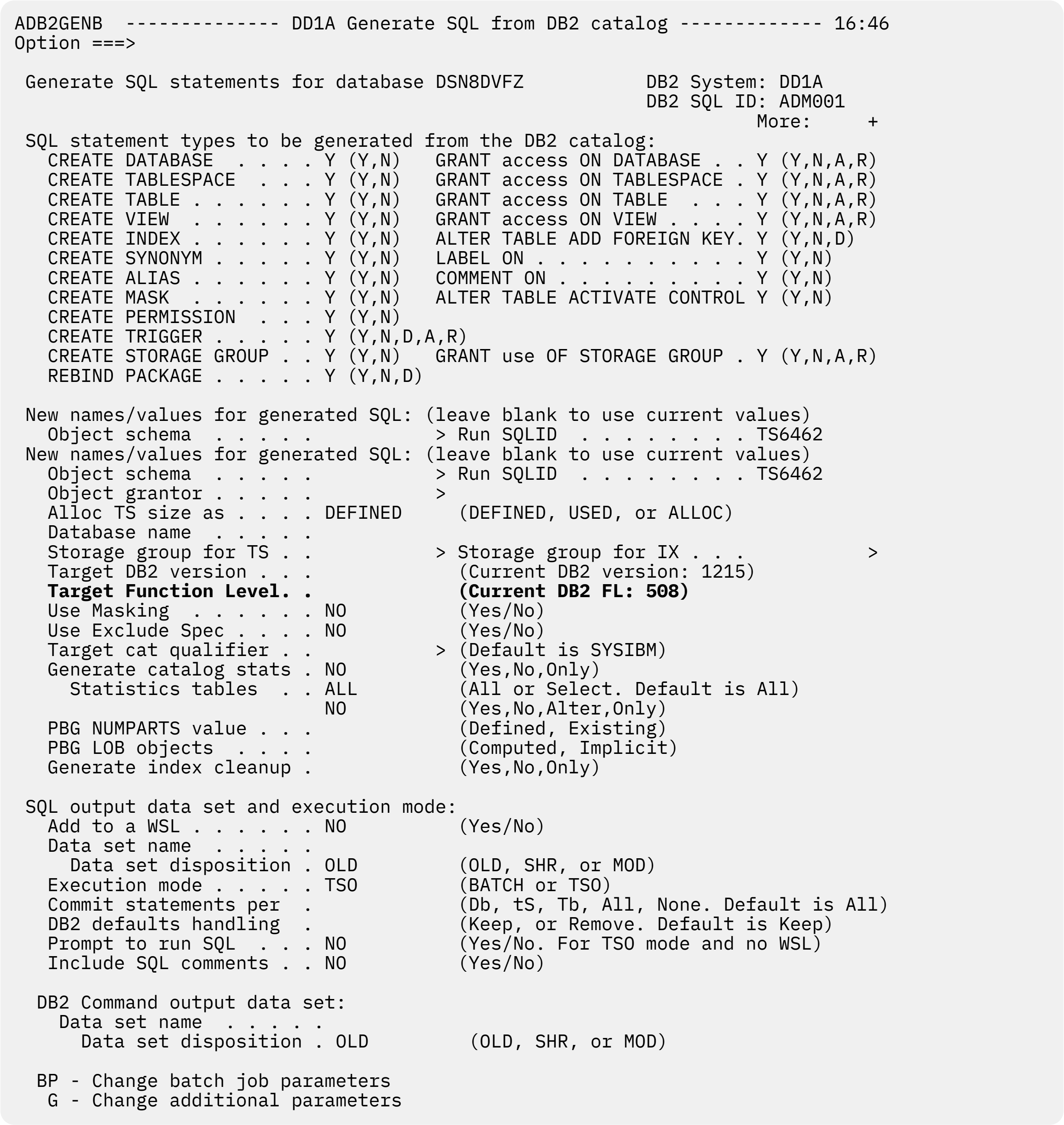
Restriction: You cannot set the APPLCOMPAT function level to a value that is greater than the function level value with which you bound ADBMAIN.

Target function level

The function level for generated DDL statements. Db2 Admin Tool generates requested DDL statements based on the syntax requirements for the target function level.

The default value is the current Db2 function level. However, you can specify any function level that is equal to or lower than the current Db2 function level. You can specify this value in one of the following places:

- The Change Management (CM) batch parameter "TGT_DB2FL" on page 727
- The ADB2RE stored procedure option TGTFL for generating statements
- On the **Generate SQL from DB2 catalog (ADB2GENB)** panel, in the **Target Function Level** field:



```
ADB2GENB  -------------- DD1A Generate SQL from DB2 catalog ------------ 16:46
Option ===>

 Generate SQL statements for database DSN8DVFZ           DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
                                                                   More:     +
 SQL statement types to be generated from the DB2 catalog:
   CREATE DATABASE  . . . . Y (Y,N)   GRANT access ON DATABASE . . Y (Y,N,A,R)
   CREATE TABLESPACE  . . . Y (Y,N)   GRANT access ON TABLESPACE . Y (Y,N,A,R)
   CREATE TABLE . . . . . . Y (Y,N)   GRANT access ON TABLE  . . . Y (Y,N,A,R)
   CREATE VIEW  . . . . . . Y (Y,N)   GRANT access ON VIEW . . . . Y (Y,N,A,R)
   CREATE INDEX . . . . . . Y (Y,N)   ALTER TABLE ADD FOREIGN KEY. Y (Y,N,D)
   CREATE SYNONYM . . . . . Y (Y,N)   LABEL ON . . . . . . . . . . Y (Y,N)
   CREATE ALIAS . . . . . . Y (Y,N)   COMMENT ON . . . . . . . . . Y (Y,N)
   CREATE MASK  . . . . . . Y (Y,N)   ALTER TABLE ACTIVATE CONTROL Y (Y,N)
   CREATE PERMISSION  . . . Y (Y,N)
   CREATE TRIGGER . . . . . Y (Y,N,D,A,R)
   CREATE STORAGE GROUP . . Y (Y,N)   GRANT use OF STORAGE GROUP . Y (Y,N,A,R)
   REBIND PACKAGE . . . . . Y (Y,N,D)

 New names/values for generated SQL: (leave blank to use current values)
   Object schema  . . . . .           > Run SQLID  . . . . . . . . TS6462
 New names/values for generated SQL: (leave blank to use current values)
   Object schema  . . . . .           > Run SQLID  . . . . . . . . TS6462
   Object grantor . . . . .           >
   Alloc TS size as . . . . DEFINED     (DEFINED, USED, or ALLOC)
   Database name  . . . . .
   Storage group for TS . .           > Storage group for IX . . .           >
   Target DB2 version . . .             (Current DB2 version: 1215)
   Target Function Level. .             (Current DB2 FL: 508)
   Use Masking  . . . . . . NO          (Yes/No)
   Use Exclude Spec . . . . NO          (Yes/No)
   Target cat qualifier . .           > (Default is SYSIBM)
   Generate catalog stats . NO          (Yes,No,Only)
     Statistics tables  . . ALL         (All or Select. Default is All)
                            NO          (Yes,No,Alter,Only)
   PBG NUMPARTS value . . .             (Defined, Existing)
   PBG LOB objects  . . . .             (Computed, Implicit)
   Generate index cleanup .             (Yes,No,Only)

 SQL output data set and execution mode:
   Add to a WSL . . . . . . NO          (Yes/No)
   Data set name  . . . . .
     Data set disposition . OLD         (OLD, SHR, or MOD)
   Execution mode . . . . . TSO         (BATCH or TSO)
   Commit statements per  .             (Db, tS, Tb, All, None. Default is All)
   DB2 defaults handling  .             (Keep, or Remove. Default is Keep)
   Prompt to run SQL  . . . NO          (Yes/No. For TSO mode and no WSL)
   Include SQL comments . . NO          (Yes/No)

  DB2 Command output data set:
    Data set name  . . . . .
      Data set disposition . OLD         (OLD, SHR, or MOD)

  BP - Change batch job parameters
   G - Change additional parameters
```

Figure 8. Target function level on the ***Generate SQL from DB2 catalog (ADB2GENB)*** *panel*

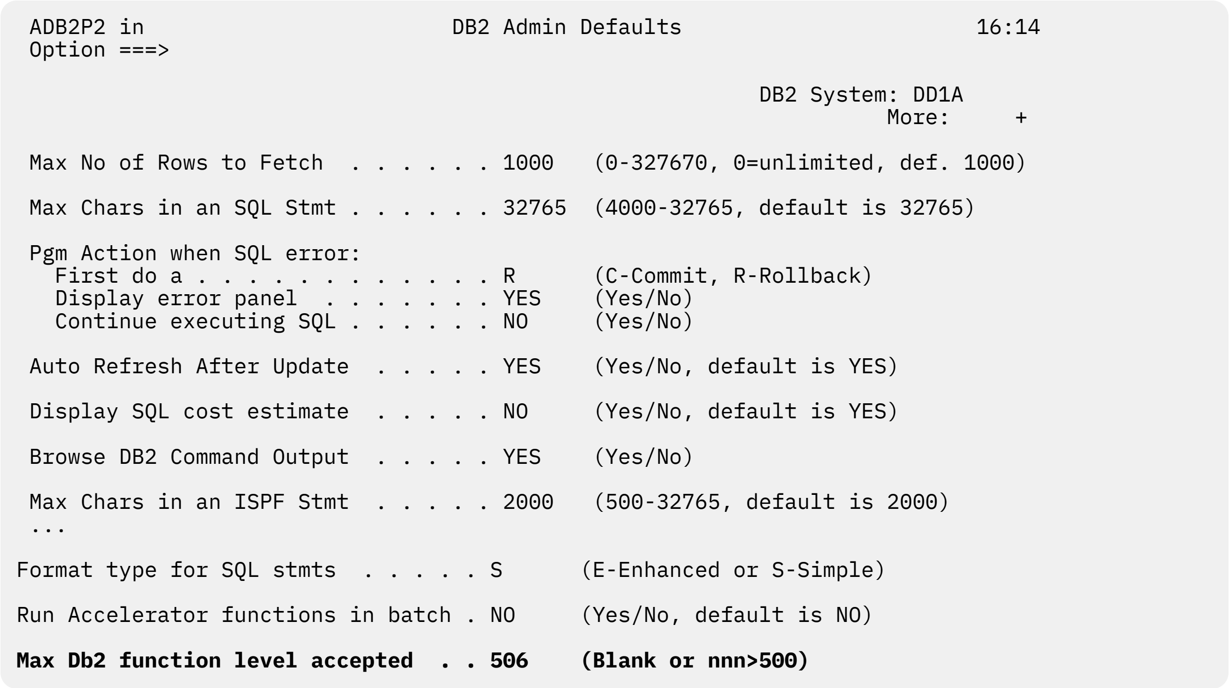
Tip: For instructions on how to navigate to this panel, see Generating SQL to re-create a Db2 object.

Maximum Db2 function level accepted

The maximum function level on which you want to allow Db2 Admin Tool to run, even if that function level is not tolerated or supported.

By default, no value is defined. You can specify this value in one of the following places:

- On the **Db2 Function Level Confirmation (ADB2CONF)** panel when you start Db2 Admin Tool. See "Db2 function level confirmation when starting Db2 Admin Tool" on page 70.
- The Change Management (CM) batch parameter "ACCEPT_FL" on page 665
- The ADB2RE stored procedure option ACCEPT FL for generating statements
- On the **Admin Defaults (ADB2P2)** panel in the **Max Db2 function level accepted** field:



```
 ADB2P2 in                        DB2 Admin Defaults                       16:14
 Option ===>

                                                          DB2 System: DD1A
                                                                    More:     +

 Max No of Rows to Fetch  . . . . . . 1000   (0-327670, 0=unlimited, def. 1000)

 Max Chars in an SQL Stmt . . . . . . 32765  (4000-32765, default is 32765)

 Pgm Action when SQL error:
   First do a . . . . . . . . . . . . R      (C-Commit, R-Rollback)
   Display error panel  . . . . . . . YES    (Yes/No)
   Continue executing SQL . . . . . . NO     (Yes/No)

 Auto Refresh After Update  . . . . . YES    (Yes/No, default is YES)

 Display SQL cost estimate  . . . . . NO     (Yes/No, default is YES)

 Browse DB2 Command Output  . . . . . YES    (Yes/No)

 Max Chars in an ISPF Stmt  . . . . . 2000   (500-32765, default is 2000)
 ...
Format type for SQL stmts  . . . . . S      (E-Enhanced or S-Simple)

Run Accelerator functions in batch . NO     (Yes/No, default is NO)

Max Db2 function level accepted  . . 506    (Blank or nnn>500)
```

Figure 9. Maximum function level accepted on the ***Admin Defaults (ADB2P2)*** *panel*

Tip: To navigate to this panel, specify `P.2` on the **DB2 Administration Menu (ADB2)** panel.

Scenarios for setting function level values

You want to recreate an object without attributes that are new in Db2 12

Suppose that your Db2 function level is 504. You want to process a change that drops and recreates a table space that was originally defined with PAGENUM in the catalog. However, you do not want to use PAGENUM when recreating the table space. (PAGENUM is a new attribute that is available in function level 500.) In this case you want to set the target function level to 100 by performing the following actions:

- **For CM batch jobs:** specify `TGT_DB2FL = '100'` as shown in the following example:

```
//*
//ANALYZE  EXEC GOCCM,SSID=DC1A,PLAN=TS5764P
//PARMS DD *
 SYMBOL_NAME = '&SCENARIO.' SYMBOL_VALUE =
'A1638B01'
 ACTION_ANALYZE_CHANGE = 'Y'
 CHANGE_NAME = '&SCENARIO..&JOBNAME.'
 CHANGE_COMMENT = 'CHANGE1'
 EXISTING_DATA_SET_ACTION='REPLACE'
 ACCEPT_FL = '504'
 TGT_DB2FL = '100'
```

- **For GEN:**

1. On the **Table Spaces (ADB21S)** panel, specify the GEN line command for the table space.
2. On the **Generate SQL from DB2 catalog (ADB2GENB)** panel, specify the following values and press Enter:
   – **Target Function Level** = 100
   – **Execution mode** = BATCH

The generated batch JCL has `ACCEPT_FL='504'` and `TGTFL='100'`:

```
//IN       DD *
  DB2SYS   = 'DC1A',
  DB2ALOC  = '',
  DB2SERV  = 'RS22DC1A',
  DB2AUTH  = 'TS5764',
  DB2REL   = '1215',
  GENSG    = 'N',
  GENDB    = '',
  GENTS    = 'Y',
  GENTABLE = 'Y',
  GENVIEW  = 'N',
  GENINDEX = 'Y',
  GENSYN   = 'N',
  GENALIAS = 'N',
  GENLABEL = 'N',
  GENCOMM  = 'N',
  GENRELS  = 'N',
  GENTRIG  = 'Y',
  ACTVCNTL = 'N',
  PENDCHGS = 'Y',
  GENMASK  = 'N',
  GENPERM  = 'N',
  GRANTDB  = '',
  GRANTTS  = 'Y',
  GRANTTAB = 'N',
  GRANTVW  = 'N',
  GRANTSG  = 'N',
  IMPLDB   = '',
  IMPLTS   = '',
  NEWDB    = '',
  NEWTSSG  = '',
  NEWIXSG  = '',
  NEWSQLID = '',
  NEWGRANTOR = '',
  SPCALLOC = 'DEFINED',
  ROLEWARN = 'Y',
  GETDB2ZP = 'Y',
  ACCEPT_FL = '504',
  TGTFL    = '100',
  TGTDB2   = '';
  DB='DBXXEE01', TS='TUXXEE01';
/*
```

The resulting DDL for CREATE TABLESPACE does not include the PAGENUM option.

Your subsystem is at FL 500 but a higher Db2 function level is specified in the ISPF profile.

In this case, the profile might have been set up with a different subsystem. To set the Db2 function level value to FL 500:

1. From the main **DB2 Administration Menu (ADB2)** panel, specify **P.CH**.
2. On the **Options for Change Functions (ADB2PCO)** panel, specify 500 in the **DB2 function level** field.

Your subsystem is at FL 505 but you have not yet applied maintenance to Db2 Admin Tool to support FL 505

If you are not using any FL 505 enhancements, Db2 Admin Tool will run as usual. However, you must specify 505 as the maximum Db2 function level accepted. When you start Db2 Admin Tool, specify option 3 on the **Db2 Function Level Confirmation (ADB2CONF)** panel. If you are running the ADB2RE stored procedure to generate SQL or running CM batch, add the following parameter to your jobs: `ACCEPT_FL = '505'`.

Related reference

“CM batch parameter definitions” on page 664

You can use Change Management (CM) batch interface parameters to control Change Management (CM) actions and settings.

Related information

Adopting new capabilities in Db2 12 continuous delivery (Db2 12 for z/OS)

Db2 12 function levels (Db2 12 for z/OS)

What does Db2 Admin Tool do?

IBM Db2 Administration Tool for z/OS, also referred to as Db2 Admin Tool, helps you with the day-to-day tasks that are associated with managing Db2 environments efficiently and effectively.

Db2 Admin Tool simplifies the complex tasks that are associated with safely managing Db2 objects and schema throughout the application lifecycle with the least possible impact to availability. The key attributes of Db2 Admin Tool include the following:

- Enables quick and easy navigation through the Db2 catalog
- Builds and executes dynamic SQL statements without requiring you to know the exact SQL syntax
- Manages and tracks changes that are made to Db2 object definitions, resolving any potential conflicts before execution
- Helps build Db2 commands to execute against databases and tables
- Builds and executes utility jobs, enabling use of LISTDEFs and TEMPLATEs for increased productivity
- Enables you to create, alter, migrate, drop and reverse engineer Db2 objects

The easy-to-use comprehensive features of Db2 Admin Tool can increase your productivity and increase the reliability of your Db2 objects:

Object management

- Provides in-depth Db2 catalog navigation, which can minimize the time that is required to review the catalog. Objects in the catalog are shown and interpreted, and relevant catalog information is presented logically. You can issue any Db2 command, including BIND, REBIND, and FREE, against selected plans and packages.

  Db2 Admin Tool presents the Db2 catalog quickly and logically:

  - Displays any object in the catalog
  - Displays related Db2 objects by using special line commands
  - Interprets catalog information
  - Displays the authorization for objects
  - Displays the static SQL statements from application plans and packages
  - Displays the DDL for existing views
  - Runs on one of multiple copies of the Db2 system catalog
- Integrated with Db2 utilities to simplify the creation of Db2 utility jobs. JCL can be generated for Db2 utilities and can be executed. The use of LISTDEFs and TEMPLATEs is also supported.
- Enables tasks such as alter, create, drop and migrate of Db2 objects
- Allows reverse engineering of Db2 objects
- Supports Db2 predictive governing
- Enables you to alter the definition of a Db2 table
- Enables you to request the Prompt function, which prompts you before a statement is executed

Security management

- Displays authorizations that have been granted on any type of Db2 object, and enables you to REVOKE these authorizations or GRANT new authorizations
- Provides REVOKE impact analysis to prevent inadvertent data loss when you revoke authorities
- Displays the list of secondary authorization IDs and manages SQL IDs

Performance management

- Allows complex performance and space queries

- Contains a built-in EXPLAIN function that allows you to EXPLAIN a query, and provides an interpretation of the PLAN_TABLE output into an easy-to-understand format
- Comes with a set of performance health check catalog queries
- Enables you to perform space-related functions such as resizing page sets, lets you move page sets to and from STOGROUP- and VCAT-defined space, and helps you estimate space allocations for new table spaces and indexes
- Enables you to create and manage work statement lists (WSLs) and run the WSL as a batch job
- Enables you to dynamically manage system parameters

Change management

- Allows you to manage and track changes to Db2 objects
- Allows you to register changes to multiple target environments
- Allows groups of users to collaborate to build changes by managing information through a series of Db2 tables
- Provides a convenient audit trail that can be used to determine the status of objects that are being changed and where those changes were deployed
- Allows you to recover changes and restore database objects to their previous state

System management

- Allows you to display and cancel threads; display and alter buffer pools; display, start, and stop Db2 traces; and set and display the logs
- Performs various system administration functions, such as updating RLIMITs and managing DDF tables
- Provides a convenient way to administer RLF and DDF tables
- Manages stored procedure operations, such as creating, displaying or altering stored procedures, issuing the Db2 START and STOP STORED PROCEDURE command, and showing statistics for stored procedures that are accessed by Db2 applications
- Displays current dynamic DSNZPARMs change parameters, generates new DSNZPARM modules with changes, and activates those changes in Db2

Application management

- Builds and executes dynamic SQL statements without requiring you to know the exact SQL syntax
- Runs most Db2 utilities
- Enables you to extend existing Db2 Admin Tool applications or to rapidly develop new applications
- Allows you to work with a copy of the Db2 catalog to avoid contention and other performance problems on the actual catalog
- Accesses a remote Db2 catalog where a DDF connection exists between systems. This feature enables you to centrally manage all of your Db2 subsystems with a single Db2 Admin session.
- Allows you to execute any dynamic SQL statement through Db2 Admin Tool, or to invoke SPUFI

Database administration and change management solutions

IBM solutions help IT organizations maximize their investment in Db2 and IMS databases and address some of today's toughest IT challenges.

Database administration and change management are the core responsibilities of the DBA. If not managed correctly, database administration and change management can monopolize data center resources, waste valuable time, and result in the generation of unwanted errors.

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

- Being able to quickly and easily navigate the Db2 catalog
- Ensuring that all of the necessary steps are completed when making a change
- Managing and tracking the changes to the definitions of database objects
- Propagating changes to other database environments
- Keeping Db2 software versions current
- Managing a corrupt database

Many Db2 Tools products provide database management features that are not available in Db2 itself or that provide enhancements to capabilities that are built into Db2.

For example, Db2 Admin Tool allows you to navigate the Db2 catalog quickly and easily.

Db2 Admin Tool provides integration with other Db2 Tools products to create extra function with product-specific line commands for editing tables, analyzing the cost of SQL statements, and analyzing potential access path changes. Db2 Admin Tool offers a central, ISPF-based access point for other Db2 Tools products, such as Db2 Table Editor, Db2 SQL Performance Analyzer, and Db2 High Performance Unload .

Db2 Admin Tool is only one of several Db2 Tools products that provide enhancements to the process of database administration and change management for your databases.

The following Db2 Tools products that can assist with database administration and change management:

- IBM Db2 Object Comparison Tool for z/OS
- Db2 Storage Management Utility
- Optim Test Data Management
- Db2 Table Editor
- Db2 SQL Performance Analyzer
- Db2 High Performance Unload

Db2 Admin Tool features and benefits

The features of Db2 Admin Tool help you to efficiently and effectively manage Db2 environments.

Db2 Admin Tool features

Display the Db2 catalog tables

Db2 Admin Tool provides extensive support for displaying the Db2 catalog. The scope of information that can be displayed is described in this information.

Display any object in the Db2 catalog

You can retrieve catalog data for any Db2 data object. You can specify the data that is retrieved (for example, you might request that data be retrieved for all databases that are owned by THOMAS and that have the prefix D402).

Db2 Admin Tool retrieves catalog data by using predefined SELECT statements for the more commonly used queries. The rows that are retrieved from the catalog are displayed using the ISPF table-display service. The display panel can be the Db2 Admin Tool default panel, from which you can issue various Db2 Admin Tool line commands, or a panel that you tailor for the result of a particular SQL SELECT. In the latter case, you can use line commands to issue new SQL calls that use information from the columns of rows that have been returned.

Display related Db2 objects using line commands

You can use Db2 Admin Tool line commands to navigate the catalog. For example, from a display panel that shows databases, you can use a line command to show all table spaces in one of the databases. Then, from the table spaces panel, you could issue a line command to show authorizations for a table space or show the status of image copies for the table space.

Display catalog information

You can request detailed information about any object in the Db2 catalog. A request for details about an application plan, for example, returns information such as the plan's owner, latest bind time, and number of bytes in the base section.

Show the authorization for Db2 objects

You can retrieve information about the authorizations for all Db2 objects. From an authorization display panel, you can then grant and revoke privileges.

Display the static SQL from application plans and packages

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

Display the DDL for existing views

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

Run with multiple copies of the Db2 catalog

This function allows you to use the Db2 system catalog, one of the many copies of it, or the catalog of a remote site. You might choose to use a different copy of the catalog for each weekday, thus associating a backup with each weekday. Or this feature can allow the system administrator to work on the actual system catalog, while developers use a copy of the catalog, thereby decreasing contention for the catalog.

Execute dynamic SQL statements

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

In addition, by entering required parameters from a panel, you can execute the following SQL statements: GRANT, REVOKE, CREATE, DROP, LABEL ON, and COMMENT ON. This feature allows you to execute the statements without knowing the exact SQL syntax; Db2 Admin Tool provides guidance for the required SQL parameters.

Manage changes to Db2 objects

Use the Change Management function to manage and track the changes that you make to the definitions of your Db2 objects. You can use the Change Management function to complete all of the steps that are typically involved with changing database objects:

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

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

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

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

Issue Db2 commands against databases and table spaces

You can issue any Db2 command against any database or table space that you have selected using Db2 Admin Tool. For example, you can issue the DISPLAY, START, and STOP commands against a database.

Db2 commands are passed to the instrumentation facility interface (IFI), and the result is displayed in ISPF browse.

Run Db2 utilities

You can generate the JCL for Db2 utilities and then run them in batch, or you can include the utility statements in a work statement list to be run at another time or on another subsystem. This function applies to the utilities for storage groups, table spaces, tables, and indexes. For example, you can, generate JCL to run the COPY, REORG, and RUNSTATS utilities for a table space.

The generated JCL consists of a JOB statement, EXEC statement, and all required DD statements. When the JCL is generated, Db2 Admin Tool invokes ISPF edit, which lets you change the JCL, submit it, or copy it to another data set.

You can generate utilities using LISTDEFs and TEMPLATEs.

Issue complex queries

You can run performance and space utilization queries against a database. The data that is returned can help you to determine whether you need to:

- Run the RUNSTATS or STOSPACE utilities
- Reorganize or redesign parts of your database or indexes
- Change the locking rule for tables
- Drop an index
- Move tables to separate table spaces
- Extend the primary allocation for a table space or index
- Reduce the size of a table space

Use the EXPLAIN function

The Db2 Admin Tool EXPLAIN function supports the EXPLAIN statement and provides related support. (The EXPLAIN statement gathers information about the access path Db2 chose to process a query.) By using the EXPLAIN function you can:

- Create a plan table (PLAN_TABLE) in the wanted database and table space.
- Issue an SQL EXPLAIN statement and see the resulting rows in the plan table.
- List a plan table to look at rows from previously executed EXPLAIN statements, or rows from BIND and REBIND operations that were executed with EXPLAIN(YES) specified.

  With this function, predefined search criteria help you find rows in the plan table. Predefined search criteria exist for application plans, DBRMs, collections, and packages. You can see the access path that is chosen by Db2 to process queries, and the tables and indexes that are accessed by Db2.
- Use EXPLAIN (ONLY) to populate EXPLAIN tables but not create a package. This option allows EXPLAIN to be run when the authorization ID of the bind or rebind process does not have the privilege to execute statements in the package.
- Upgrade a plan table to the current version of Db2.
- Look at the Db2 calculated cost.
- Create and display the Db2 explain tables.
- Insert and work with Db2 optimizer hints in the plan table.

Manage SQL IDs

You can change the current Db2 SQL ID by entering a new one or by selecting one from a list of secondary SQL IDs. Db2 Admin Tool displays a list of SQL IDs that you are allowed to use. The list is created either by simulating or invoking the authorization exit in your system.

Perform system administration functions

The system administration functions that you can perform using Db2 Admin Tool include:

- Displaying threads
- Displaying and terminating utilities
- Displaying and managing traces
- Displaying and updating RLIMITs, including the predictive governing limits in Db2
- Displaying and altering buffer pools
- Displaying and setting archive log parameters and archiving the log
- Displaying Db2 system parameters and updating dynamic parameters
- For DDF (distributed data facility):
  - Starting and stopping DDF
  - Displaying and updating the communications database (CDB)
  - Displaying and canceling distributed threads
  - Displaying active locations
- Dynamically managing system parameters

Reverse engineer Db2 objects

Reverse engineering generates the SQL statements necessary to re-create a Db2 object. You can reverse engineer the Db2 objects in your database catalog.

Typical uses for the Db2 Admin Tool reverse engineering function include the following tasks:

- Extracting the DDL for an object before changes are made, so that the changes are applied to the current definition and are available for fallback purposes.
- Moving Db2 objects to another Db2 subsystem. By using the reverse engineering function (together with the table unload and load functions), objects can be moved after a few manual modifications to the generated SQL and batch jobs.

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

Use the Db2 predictive governing

You can use Db2 Admin Tool to display, insert, update, or delete predictive governing rows in the resource limit tables. Furthermore, if Db2 Admin Tool receives a predictive governing warning (SQLCODE +495) when running a dynamic SQL statement, Db2 Admin Tool ask whether the SQL statement should be executed or cancelled. If the predictive governing estimates that executing a dynamic SQL statement that was issued from Db2 Admin Tool will exceed the error limit (SQLCODE -495), Db2 Admin Tool displays an error message, and the SQL statement is not executed.

You can use predictive governing limits to prevent users from running *wild* queries on catalog tables or any other tables that aredisplayed using Db2 Admin Tool. By using predictive governing limits for the Db2 Admin Tool package, this type of query can be inhibited either by setting up a predictive governing warning or an error limit in the resource limit table.

Related information:

Limiting resources for SQL statements predictively (Db2 12 for z/OS)

Alter the Db2 table definition

You can alter the definition of a Db2 table. Permissible changes include the following tasks:

- Changing the database, table space, owner, and the name of the table
- Modifying the definitions of table columns

- Changing the sequence of the columns in the table
- Inserting and dropping columns

Migrate Db2 data to other Db2 systems

You can copy Db2 data to another Db2 system. This is a useful function if you want to create a separate Db2 test system or if you want to move a test system into production. You can also use this function to consolidate two separate database systems into one.

Extend existing Db2 Admin Tool applications or develop new applications

You can extend Db2 Admin Tool to invoke other ISPF applications that you use for Db2 Admin Tool and application development. Some applications that you might want to invoke from Db2 Admin Tool are as follows:

- Security tools
- Vendor Db2 utilities
- Storage management tools

Db2 Admin Tool also enables you to quickly build new ISPF applications for displaying and maintaining Db2 data. Some of the types of data for which you might build such applications are as follows:

- Application definition data
- Db2 performance data
- Extra security data

A sample application is included with the product to illustrate how you might use Db2 Admin Tool to create new applications.

Perform space management functions

Db2 Admin Tool enables you to perform space-related functions such as resizing page sets, moving page sets to and from STOGROUP- and VCAT-defined space, and estimating space allocations for new table spaces and indexes.

Create and run work statement lists

Db2 Admin Tool enables you to create and run work statement lists that include sets of operations. You can execute the entire set, rerun sets, or capture a set of operations that you create on one system for use on another system.

Launch installed IBM Db2 Tools that have an ISPF interface

You can invoke installed IBM Db2 tools that have an ISPF interface—directly from Db2 Admin Tool. The Db2 Admin Launchpad provides a convenient way of creating a centralized ISPF table with the names of your tools. Then, by selecting an entry in this table, you can easily start one of the tools.

Performance

Db2 Admin Tool is equipped with the following performance features:

- Db2 Admin Tool uses dynamic SQL to access the Db2 catalog, which ensures that Db2 always uses the most efficient access path to the catalog (provided RUNSTATS statistics are available for the Db2 optimizer).
- Before Db2 Admin Tool displays information, it does an SQL commit. By doing so, Db2 Admin Tool ensures that a user cannot lock the catalog for long periods of time. If an SQL error occurs, Db2 Admin Tool rolls back the unit of work before it displays any information.

- Db2 Admin Tool has a default limit of 1000 for fetching rows. This limit helps to prevent time-consuming queries. You can change the default of 1000 for an execution of Db2 Admin Tool if more rows are needed. You can set this value permanently or you can set a parameter in the **DB2 Admin Defaults (ADB2P2)** panel to reset the default value at the next startup.
- You can use Db2 resource limit facilities (RLF) to limit the amount of CPU time that a dynamic SQL statement in Db2 Admin Tool can use - either by using the reactive governor facilities of RLF or by using the predictive governing facilities.
- Db2 Admin Tool can run on a copy of the Db2 catalog. Besides improving performance, running on a copy of the catalog can reduce contention for the catalog. Db2 Admin Tool provides commands to generate jobs to create and populate copies of the Db2 catalog.

Security

Db2 Admin Tool does not expose the security of the Db2 system. All Db2 access is controlled by the existing security provisions of the Db2 system. You can optionally configure Db2 Admin Tool to allow users to execute DDL generated to re-create views that are dropped as a result of dropping other objects. The user can execute this DDL even if they do not have the direct authority. This is done by using *auth-switching* and has its own safeguards to ensure the DDL is not changed from that generated by Db2 Admin Tool. A user must have access to a RACF (or equivalent) profile to use auth-switching.

Db2 Admin Tool benefits

This section describes a few of the many ways in which Db2 Admin Tool is typically used, and gives examples of specific applications.

Explore databases

Db2 Admin Tool lets you quickly navigate the Db2 catalog and display tables, table columns, and indexes. If you are authorized by Db2, you can also display the content of tables either by doing a simple list of the table or by building SQL statements and executing them against a table.

You can use the Db2 Admin Tool functions to explore unknown databases rapidly or get a quick overview of a database. None of these uses require that you remember the exact syntax of Db2 commands or SQL statements.

Determine and correct problems

You can use Db2 Admin Tool to identify and fix problems with your databases. With its ability to navigate the catalog and use Db2 commands on objects, Db2 Admin Tool can help you discover, analyze, and fix database problems in a more user-friendly fashion than is available with native Db2.

Develop small applications

You can use Db2 Admin Tool to rapidly develop small applications. As you become familiar with the tool, you might find the time that it takes to develop small Db2 Admin Tool dialogs is greatly reduced.

Examples

- If you have a tool at your installation that manipulates Db2 tables, you can develop your own line command to access it from the Db2 Admin Tool panel that displays tables (implementing the line command as an SMP usermod). Then you can invoke the table tool as a natural follow-on to using Db2 Admin Tool.
- Perhaps you want to generate more DECLARE statements for a PL/I table than is possible with the Db2 DCLGEN tool. You can write an application to invoke DCLGEN directly from the Db2 Admin Tool panel that displays tables. You can also modify the output you receive from DCLGEN to, for example, meet your installation's standards and requirements.

- You might want to build prototypes of SQL SELECT statements. You can build the statements, test them and, when you are satisfied with them, copy the statements to a data set to include in your application program.
- Db2 Admin Tool can help you maintain any Db2 tables that you use for installation standards and special requirements. You can use Db2 Admin Tool to develop a small application that describes all of the applications that you have in the system. Or you can use it to display existing tables that, for example, contain information about Db2 plan performance or batch job execution statistics.

Copy tables from one Db2 system to another

You can use the table utilities that Db2 Admin Tool generates to copy tables from one Db2 system to another. You need to make a few modifications to the generated JCL.

Start Db2 Tools

You can invoke installed IBM Db2 tools that have an ISPF interface directly from Db2 Admin Tool. Db2 Admin Tool guides you through the process of creating a central table with the names of your Db2 utilities. After this table is created, you can select an entry in it to start one of the Db2 tools.

Multi-factor authentication support

The primary user interface to Db2 Admin Tool and Object Comparison Tool is through its ISPF application, which has multi-factor authentication (MFA) support through TSO logon processing. All authentication is done before Db2 Admin Tool is invoked and no additional MFA processing is necessary.

All batch jobs created by Db2 Admin Tool and submitted for execution inherit the ID that has already been authenticated by the submitter. The Db2 services that are used by Db2 Admin Tool are invoked using the standard attachment facilities. For the stored procedures that can be accessed by REST services, the authentication is done by Db2.

Related information

TSO/E (IBM Z Multi-Factor Authentication 2.1.0)

Multi-Factor Authentication for z/OS (z/OS 3.1.0)

Service updates and support information

Service updates and support information for Db2 Admin Tool, including fixes and updates, are available from the web.

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

https://www.ibm.com/mysupport

Product documentation and updates

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

The Db2 Admin Tool documentation is available in the following two formats:

Topics in IBM Documentation (IBM Docs)

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

You can find IBM Db2 Administration Tool for z/OS in IBM Documentation at https://www.ibm.com/docs/en/db2admintool.

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 Administration 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/SSAUZ9_12.1.0/pdf/adbugl14.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.

Programming interface information

The documentation for IBM Db2 Administration Tool for z/OS is defined in one of the following ways:

NOT programming interface information (NOTPI)

NOT programming interface information is any documented information that **IS NOT** a programming interface but is provided for specialized tasks, such as diagnosis, modifying, monitoring, repairing, tailoring, and tuning. Any text in this category is labeled with the following tags:

```
+---------------------NOT programming interface information----------------------+
```

```
+------------------End of NOT programming interface information------------------+
```

Other information about the product

Other information about the product is information that is not NOTPI.

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 install Db2 Admin Tool for the first time or when you migrate to a new release of Db2 Admin Tool, you must customize the configuration. After the installation or migration, you might later need to recustomize Db2 Admin Tool for certain situations. For example, when applying maintenance, the instructions might direct you to recustomize Db2 Admin Tool.

To customize or recustomize Db2 Admin 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 Db2 Admin Tool. For detailed information about how to use TCz, see IBM Tools Customizer for z/OS 1.1.0 .

Customization checklist for Db2 Admin Tool

The following checklist describes each significant customization step. Use this checklist to guide you through the entire customization process for Db2 Admin Tool.

Tip: For help with IBM Tools Customizer for z/OS terminology, see Tools Customizer terminology and data sets (IBM Tools Customizer for z/OS 1.1).

Task	Link to detailed instructions
Verify hardware requirements	
Ensure that you deploy Db2 Admin Tool on a z-series processor that is capable of running z/OS Version 1 Release 12 or higher.	None
Very software requirements	
Verify that your environment meets the minimum software requirements.	“Software requirements for Db2 Admin Tool 12.1” on page 89
Verify SMP/E installation	
Verify that Db2 Admin Tool is installed. SMP/E installation instructions are in the program directory.	Program Directory for Db2 Admin Tool V12.1 (GI13-4642)
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)
Optional: APF authorize SADBLINK	
Several programs and TSO commands must be APF authorized to use them within Db2 Admin Tool. Therefore, ensure that the SADBLINK data set is APF authorized. Alternatively, you can later copy the appropriate programs to an APF-authorized library (as described under Required in some cases: Update the APF Authorization table later in this table).	None
Gather any data set names and subsystem names	

Task	Link to detailed instructions
Record the data set names that you will need during the customization process. Record the names of all Db2 subsystems on which you want to run Db2 Admin Tool.	"Data sets used by Tools Customizer" on page 90
Customize products that will be launched from Db2 Admin Tool	
If you plan to enable any of the following products to be launched from Db2 Admin Tool, customize these products before you customize Db2 Admin Tool: • IBM Db2 Object Comparison Tool for z/OS • Db2 Cloning Tool • Db2 High Performance Unload • Db2 Table Editor	Customizing Db2 Object Comparison Tool (IBM Db2 Object Comparison Tool for z/OS 12.1.0) Customizing Db2 Cloning Tool (IBM Db2 Cloning Tool for z/OS 3.2.0) Db2 HPU customization (IBM Db2 High Performance Unload for z/OS 5.1.0) Customizing Db2 Table Editor (IBM Db2 Table Editor 4.5)
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 97
Customize Db2 Admin Tool	
Complete the steps in the appropriate customization roadmap based on the type of customization that you are performing. If you are recustomizing Db2 Admin Tool, because you migrated to a new Db2 version, mode, or function level, complete the steps for updating Db2 Admin Tool instead.	"Roadmap: Customizing Db2 Admin Tool for the first time" on page 91 "Roadmap: Migrating to Db2 Admin Tool 12.1 from a previous release" on page 93 "Roadmap: Recustomizing Db2 Admin Tool" on page 96 "Updating Db2 Admin Tool after migrating to a new Db2 version, mode, or function level" on page 98
Required in some cases: Update the APF Authorization table	
Complete this step if the SADBLINK library is not APF authorized or you plan to use HPU. In these cases, copy the ADB2ATH and ADB2UTIL programs and optionally INZUTILB (for Db2 High Performance Unload) to an APF-authorized library.	"Updating the APF authorization table" on page 110
Optional: Verify activation of limited functionality	
Verify that Db2 Admin Tool is operational.	"Verifying activation of limited functionality" on page 111
Optional: Specify naming conventions	
If you want Db2 Admin Tool to use your local naming conventions for items such as data sets, utility IDs (UIDs), and plan names, update member SADBSLIB(ADB2UCUS).	"Specifying naming conventions" on page 114
Optional: Tailor the Db2 Admin Tool Launchpad	

Task	Link to detailed instructions
The Db2 Admin Tool Launchpad enables you to launch all installed IBM Db2 tools that have an ISPF interface directly from a centralized panel.	“Creating the Launchpad table” on page 127
Optional: Grant SELECT access on catalog tables	
If you plan to make Db2 Admin Tool available to a large number of users, considering running a TCz job to authorize specific IDs to see the catalog.	“Granting SELECT access on catalog tables” on page 116
Optional: Optimize ADMIN_INFO_SYSPARM and DSNZPARM settings	
To optimize performance, verify that the ADMIN_INFO_SYSPARM stored procedure is operational and that the DSNZPARM STORTIME(DSN6SYSP) parameter is set to a reasonable time.	“Optimizing ADMIN_INFO_SYSPARM and DSNZPARM settings for GEN and DDL” on page 116
Optional: Run the RUNSTATS utility	
Run the RUNSTATS utility on the Db2 catalog. This action is recommended to optimize performance.	None
Optional: Define reverse engineering stored procedures	
If you want to apply the reverse engineering functionality in Db2 Admin Tool to additional software products, define stored procedures ADB2RE and ADBGDDL.	“Defining the provided stored procedures ” on page 117
Optional: Enable Db2 Admin Tool distributed support	
To use Db2 Admin Tool on remote Db2 systems, fully enable distributed support.	“Enabling distributed support” on page 118
Optional: Improve performance when making Db2 Admin Tool available to users	
To reduce Db2 Admin Tool start-up time, use one of the specified methods for improving performance.	“Improving performance when making Db2 Admin Tool available to users” on page 119
Optional: Set global variables	
To set Db2 Admin Tool variables site-wide for all users, use the provided sample invocation exit.	“Setting global variables for Db2 Admin Tool” on page 125
Optional: Tailor Db2 Admin Tool Authorization Switching	
Authorization switching is deprecated in Db2 Admin Tool. If you still want to use this functionality, complete the setup by following the instructions.	“Tailoring Authorization Switching” on page 119

Preparing to customize Db2 Admin Tool

Before you use TCz to customize Db2 Admin Tool, review the software requirements and gather the information that you will need.

Software requirements for Db2 Admin Tool 12.1

Prior to beginning the customization process for Db2 Admin Tool, ensure that your environment meets all software requirements.

Db2 Admin Tool 12.1 requires the following software:

- z/OS 1.12 (5694-A01) or later

  To run Db2 Admin Tool on z/OS 2.5, you must have PTF UJ07166 applied.
- IBM System Modification Program Extended (SMP/E) for z/OS, 3.5 or higher
- One of the following supported versions of Db2 for z/OS:
  - Db2 12 for z/OS (5650-DB2®)
  - Db2 Value Unit Edition 12.1 (5770-AF3)
  - Db2 11 for z/OS (5615-DB2)
  - Db2 Value Unit Edition 11.1 (5697-P43)

Optionally, if you plan to use the following additional features and tools, ensure that you have the specified tool and version installed:

- IBM Db2 Object Comparison Tool for z/OS 12.1 (5655-DC2) (Required if you plan to use the Change Management function in Db2 Admin Tool)
- IBM Db2 Cloning Tool for z/OS 3.1 (5655-N15)
- IBM Db2 High Performance Unload for z/OS 4.2 (5655-AA1)
- IBM Db2 Table Editor for z/OS 4.5 (5697-G65)

Memory recommendations: Because certain functions of Db2 Administration Tool keep 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.

Related information

Db2 Object Comparison Tool overview (IBM Db2 Object Comparison Tool for z/OS 12.1.0)

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 Admin Tool

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

About this task

For every Db2 subsystem on which you want to use Db2 Admin Tool, you must run TCz to customize Db2 Admin Tool.

Recommendation: Never modify the SMP/E target libraries or run jobs from the target libraries. Typically, you create run time libraries based on these target libraries. ADBL CLIST also has a mechanism to allow you to integrate modifications into a set of separate user libraries that are concatenated to the run time libraries. This mechanism ensures that you never lose your modifications.

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

Roadmap: Customizing Db2 Admin Tool for the first time

When you install Db2 Admin Tool for the first time, you must customize the configuration by using Tools Customizer.

Complete the steps in the following table to customize Db2 Admin Tool for the first time.

Tips:

- For multiple-LPAR environments, determine your customization strategy first: "Using Tools Customizer in a multiple-LPAR environment" on page 97.
- For guidance on any input fields in TCz, position your cursor on the input field and press F1 (Help).

Table 4. Steps for customizing Db2 Admin Tool for the first time

Step	Procedure	Links to more information

Start Tools Customizer.	1. Edit the CCQTCZ member in the *hlq*.TCZ110.SCCQEXEC data set.
	2. Locate `TCZHLQ="<TCz HLQ>"`.
	3. Change `"<TCz HLQ>"` to the high-level qualifier of your Tools Customizer EXEC data set, as shown in the following example:
	`TCZHLQ="hlq.TCZ110"`
	4. Save your changes.
	5. On the ISPF Command shell panel, issue the following command:
	`EX 'hlq.TCZ110.SCCQEXEC(CCQTCZ)'`
		• Starting Tools Customizer (IBM Tools Customizer for z/OS 1.1)
		• "Data sets used by Tools Customizer" on page 90

Table 4. Steps for customizing Db2 Admin Tool for the first time (continued)

Step	Procedure	Links to more information
Modify Tools Customizer settings.	1. On the **IBM Tools Customizer for z/OS (CCQPHME)** panel, specify option `0` (**User settings for Tools Customizer**), and press Enter 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 Db2 Admin Tool metadata library.	1. On the **IBM Tools Customizer for z/OS (CCQPHME)** panel, specify option 1 (**Customize a product**), and press Enter. 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.ADB2PAR.SADBDENU`. The default name of the product metadata library is *high_level_qualifier*.SADBDENU, where *high_level_qualifier* is all of the segments of the data set name except the lowest-level qualifier.	Specifying the metadata library for the product or pack to customize (IBM Tools Customizer for z/OS 1.1)
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.	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 (CCQPPRD)** panel, specify your parameter values. Required parameters are indicated by an asterisk (*). 3. Press Enter to save and exit.	"Defining Db2 Admin Tool parameters" on page 100

Table 4. Steps for customizing Db2 Admin Tool for the first time (continued)

Step	Procedure	Links to more information
Define LPAR parameters.	1. On the **Customizer Workplace (CCQPWRK)** panel, specify the E line command against the **LPAR parameters** field. 2. On the **LPAR Parameters (CCQPLPR)** panel, specify your parameter values. 3. Press Enter to save and exit.	"Defining LPAR parameters" on page 104
Edit the Db2 entry.	1. On the **Customizer Workplace (CCQPWRK)** panel, specify the E line command against the new Db2 entry to edit the parameters. 2. On the **DB2 Parameters (CCQPDB2)** panel, specify your parameter values. Required parameters are indicated by an asterisk (*). **Tip:** Some of the parameters on the CCQPDB2 panel are identical to parameters on the CCQPPRD panel. If you leave these parameters blank on the CCQPDB2 panel, Tools Customizer uses the values specified on the CCQPPRD panel. If you use unique values for specific Db2 entries, specify these values on the CCQPDB2 panel. For example, if five Db2 subsystems use the STD Db2 security exit, specify STD on the CCQPPRD panel and leave the field blank on the CCQPDB2 panel for each subsystem. 3. Press Enter to save and exit.	"Defining Db2 parameters" on page 106
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)
Submit the jobs.	On the **Finish Product Customization (CCQPCST)** panel, issue the E line command against the *ab*CUST*xy* member. **Important:** These are the minimum jobs to be submitted.	"Submitting the customization jobs" on page 107
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 97

Roadmap: Migrating to Db2 Admin Tool 12.1 from a previous release

When you migrate to a new release of Db2 Admin Tool from a previous release, you must customize the configuration by using Tools Customizer (TCz).

Complete the steps in the following table to migrate to Db2 Admin Tool 12.1 from a previous release.

Tips:

- For multiple-LPAR environments, determine your customization strategy first: "Using Tools Customizer in a multiple-LPAR environment" on page 97.
- For guidance on any input fields in TCz, position your cursor on the input field and press F1 (Help).

Table 5. Steps for migrating to Db2 Admin Tool 12.1 from a previous release

Step	Procedure	Links to more information
Start Tools Customizer.	1. On the ISPF Command shell panel, issue the following command: `EX 'hlq.TCZ110.SCCQEXEC(CCQTCZ)'`	Starting Tools Customizer (IBM Tools Customizer for z/OS 1.1)
Specify the Db2 Admin 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, specify the Db2 Admin Tool metadata library in the **Product or pack metadata library** field, such as `DMTOOL.ADB2PAR.SADBDENU`. The default name of the product metadata library is *high_level_qualifier*.SADBDENU, where *high_level_qualifier* is all of the segments of the data set name except the lowest-level qualifier.	Specifying the metadata library for the product or pack to customize (IBM Tools Customizer for z/OS 1.1)
Run the Discover EXEC.	1. On the **Customizer Workplace (CCQPWRK)** panel, issue the DISCOVER command, and press Enter. 2. On the **Discover Customized Product Information (CCQPDSC)** panel, specify values for the following required input fields: • **Source Customized table library** (`DMTOOL.ADBB1PAR.SADBTLIB`) • **Target Customized table library** (`DMTOOL.ADBC1PAR.SADBTLIB`) 3. Issue the RUN command. Upon completion, the DISCOVER process populates all Tools Customizer input fields from the previous customization to the current customization. Specifically, the DISCOVER process automatically copies the settings from the customization table ADBTPARM and populates the input fields on the **Product Parameters (CCQPPRD)** panel and the **DB2 Parameters (CCQPDB2)** panel. If you use Launchpad, the DISCOVER process also ensures that the Launchpad table ADBDMT is discovered and copied from the specified site-specific source data set to the target data set. If DISCOVER is not used during an Db2 Admin Tool upgrade, the Launchpad panel (if used) might be missing Launchpad product settings that were established before the upgrade.	"Discovering Db2 Admin Tool information automatically" on page 99

Table 5. Steps for migrating to Db2 Admin Tool 12.1 from a previous release (continued)

Step	Procedure	Links to more information
Define product parameters.	1. On the **Customizer Workplace (CCQPWRK)** panel, specify the E line command against the **Product parameters** field. 2. Confirm that the DISCOVER process populated the input fields accordingly. Verify the following input fields to ensure that the current libraries for Db2 Admin Tool and Object Comparison Tool are specified: • **Admin Tool/OC CLIST** • **Admin Tool DBRM** • **Admin Tool/OC EXEC** • **Admin Tool/OC Load** • **Admin Tool/OC Message** • **Admin Tool/OC Panel** • **Admin Tool/OC Skeleton** • **Admin Tool/OC Table** • **Customized Table lib** • **Admin Tool hlq** • **Admin Tool PROCLIB**	"Defining Db2 Admin Tool parameters" on page 100
Define LPAR parameters.	1. On the CCQPWRK panel, specify the E line command against the **LPAR parameters** field. 2. Confirm that the DISCOVER process populated the input fields accordingly.	"Defining LPAR parameters" on page 104
Edit the new Db2 entry.	1. On the **Customizer Workplace (CCQPWRK)** panel, specify the E line command against the site-specific SSID to display the Db2 parameters. 2. On the **DB2 Parameters (CCQPDB2)** panel, confirm that the DISCOVER process populated the input fields accordingly.	"Defining Db2 parameters" on page 106
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)
Submit the jobs.	On the **Finish Product Customization (CCQPCST)** panel, submit the generated jobs in the order that they are displayed.	"Submitting the customization jobs" on page 107
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 97

Roadmap: Recustomizing Db2 Admin Tool

When you apply maintenance, you might need to run Tools Customizer (TCz) again to recustomize Db2 Admin Tool. This recustomization process involves changing any parameter values and generating new customization jobs.

Important: When applying Db2 Admin Tool maintenance, do not override SADBTLIB members ADBDMT and ADBTPARM and SADBSLIB member ADB2UCUS.

To recustomize Db2 Admin Tool, complete the steps in the following table.

Tips:

- For multiple-LPAR environments, determine your customization strategy: "Using Tools Customizer in a multiple-LPAR environment" on page 97.
- 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 6. Steps for recustomizing Db2 Admin Tool

Step	Procedure	Links to more information
Start Tools Customizer.	1. On the ISPF Command shell panel, issue the following command: `EX 'hlq.TCZ110.SCCQEXEC(CCQTCZ)'`	Starting Tools Customizer (IBM Tools Customizer for z/OS 1.1)
Specify the Db2 Admin 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, specify the Db2 Admin Tool metadata library in the **Product or pack metadata library** field, such as `DMTOOL.SADBDENU`. The default name of the product metadata library is *high_level_qualifier*.SADBDENU, where *high_level_qualifier* is all of the segments of the data set name except the lowest-level qualifier.	Specifying the metadata library for the product or pack to customize (IBM Tools Customizer for z/OS 1.1)

Table 6. Steps for recustomizing Db2 Admin Tool (continued)

Step	Procedure	Links to more information
Define product parameters, LPAR parameters, or Db2 parameters.	1. On the **Customizer Workplace (CCQPWRK)** panel, specify the E line command against one of the following fields, depending on which parameters you need to change, and press Enter: • **Product parameters** • **LPAR parameters** • a Db2 entry 2. Edit the specific tasks, steps, or parameters that you want to change. 3. Press Enter to save and exit.	• "Defining Db2 Admin Tool parameters" on page 100 • "Defining LPAR parameters" on page 104 • "Defining Db2 parameters" on page 106
Generate the jobs.	On the **Customizer Workplace (CCQPWRK)** panel, specify the G line command against a site-specific SSID, and press Enter.	Generating customization jobs (IBM Tools Customizer for z/OS 1.1)
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 107
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 97

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 panel (CCQPCST)**. Depending on the values of the **SSID** and **GrpAttch** columns, take the following actions for each job:

Table 7.

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

Updating Db2 Admin Tool after migrating to a new Db2 version, mode, or function level

If you migrate from one version, mode, or function level of Db2 for z/OS to another, you need to make sure that Db2 Admin Tool accounts for any Db2 catalog changes in the new version, mode, or function level.

Procedure

To update Db2 Admin Tool after migrating to a new Db2 version, mode, or function level:

1. Regenerate and run any job that was generated by Tools Customizer (TCz) and marked as new on **Finish Product Customization (CCQPCST)** panel. (The **New** column contains YES.) Usually, those jobs are ADBSETUP and ADBBIND. You need to submit these jobs on all affected Db2 subsystems. To regenerate and run these jobs, take the following steps in TCz:

   Tip: For more specific TCz instructions on any of these steps, see "Roadmap: Recustomizing Db2 Admin Tool" on page 96.

   a) **For Db2 12:** Set the following TCz parameters:

- On the **Product Parameters (CCQPPRD)** panel, in the **Convert Admin table spaces to UTS** field, specify YES to ensure that all ADB table spaces are converted to universal table spaces.
- On the **DB2 Parameters (CCQPDB2)** panel:
  - In the **Mode** field, specify CM for FL 100 or NFM for FL 500 or higher.
  - In the **APPLCOMPAT** field, specify a value according to the recommendations in "What APPLCOMPAT value should I specify for Db2 Admin Tool packages?" on page 67. If you leave this field blank, packages are bound with the default zparm value.

b) On the **Customizer Workplace (CCQPWRK)** panel, issue the G line command to regenerate the necessary job templates. TCz generates the ADBSETUP (Admin Tool Setup Task) job template and the ADBBIND (Bind Packages) and/or ADBPLANS (Bind Plans) job templates.

c) On the **Finish Product Customization (CCQPCST)** panel, issue the E line command to select job template ADBSETUP, and submit the job.

d) Return to the **Finish Product Customization (CCQPCST)** panel, issue the E line command to select job templates ADBBIND and ADBPLANS, and submit each job. You can ignore bind errors when running ADBBIND if the errors are related to the Db2 catalog tables.

2. If you defined multiple copies of the Db2 catalog before upgrading to a new release or mode, re-run the bind steps for the catalog copies that you created. See "Using previously defined copies of the Db2 catalog" on page 1054.

Discovering Db2 Admin Tool information automatically

You can use the Db2 Admin Tool Discover EXEC to discover information from a previous or current customization of Db2 Admin Tool.

About this task

The DISCOVER process automatically copies the settings from the customization table ADBTPARM and populates the input fields on the **Product Parameters (CCQPPRD)** panel. If you use Launchpad, the DISCOVER process also ensures that the Launchpad table ADBDMT is discovered and copied from the specified site-specific source data set to the target data set.

Using the Db2 Admin Tool Discover EXEC to discover information from a previous or current customization saves time and reduces errors that can occur when parameters are specified manually. For example, if the Discover EXEC is not used during an Db2 Admin Tool upgrade, the Launchpad panel (if used) might be missing Launchpad product settings that were established before the upgrade.

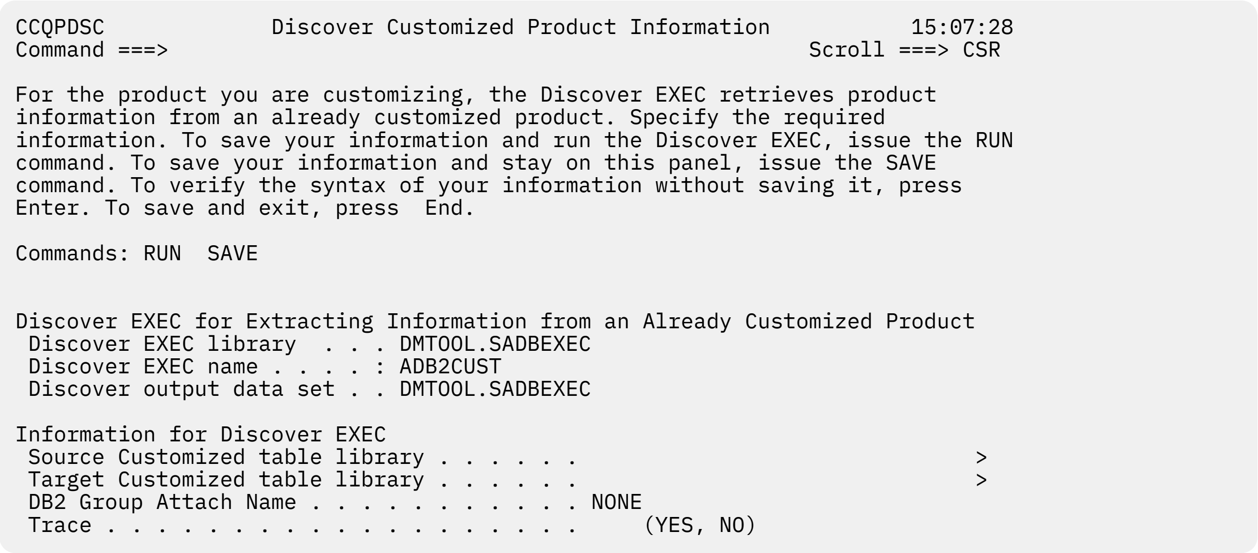
Procedure

To discover Db2 Admin Tool information automatically:

1. On the **Customizer Workplace (CCQPWRK)** panel, issue the `DISCOVER` command.

   Issuing this command is not necessary if you chose to run the Db2 Admin Tool Discover EXEC on the pop-up window after you specified the product to customize.

   The **Discover Customized Product Information (CCQPDSC)** panel is displayed:



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

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

Commands: RUN  SAVE


Discover EXEC for Extracting Information from an Already Customized Product
 Discover EXEC library  . . . DMTOOL.SADBEXEC
 Discover EXEC name . . . . : ADB2CUST
 Discover output data set . . DMTOOL.SADBEXEC

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

Figure 10. ***Discover Customized Product Information (CCQPDSC)*** *panel*

2. Run the EXEC by completing the following procedure: Discovering product or component information automatically (IBM Tools Customizer for z/OS 1.1)

Related tasks

“Defining Db2 Admin Tool parameters” on page 100
In TCz, *Db2 Admin Tool parameters* contain values that are specific to Db2 Admin Tool. If you ran the Db2 Admin Tool Discover EXEC, these parameter values were discovered for you; you need only to review them.

“Defining LPAR parameters” on page 104
In TCz, *LPAR parameters* are parameters on the local LPAR that are required to customize Db2 Admin Tool.

“Defining Db2 parameters” on page 106
In TCz, *Db2 parameters* are parameters for a Db2 entry.

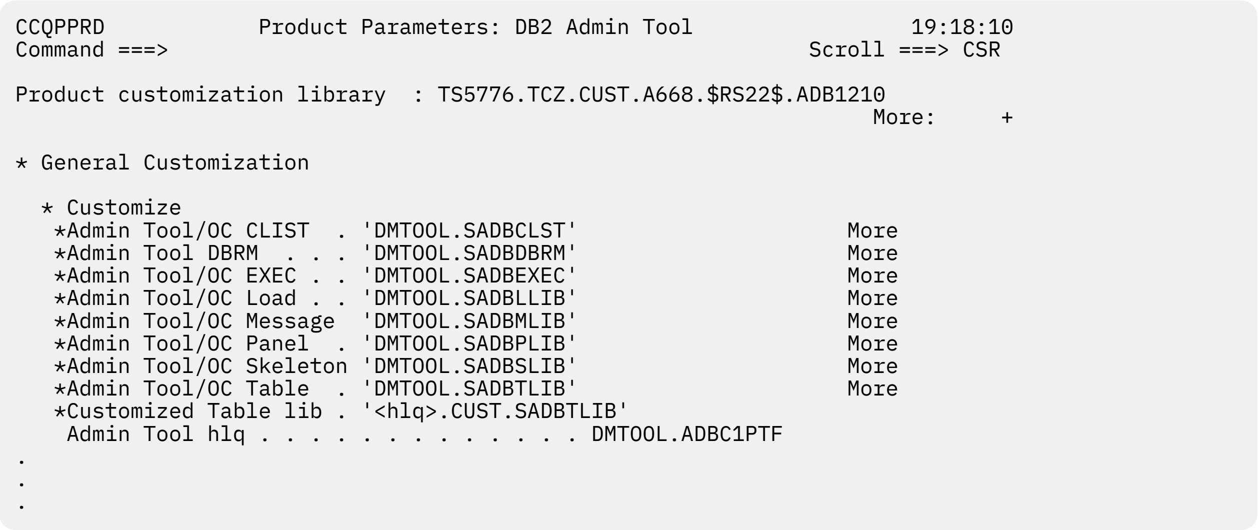
Defining Db2 Admin Tool parameters

In TCz, *Db2 Admin Tool parameters* contain values that are specific to Db2 Admin Tool. If you ran the Db2 Admin Tool Discover EXEC, these parameter values were discovered for you; you need only to review them.

Procedure

To define Db2 Admin Tool parameters:

1. On the **Customizer Workplace (CCQPWRK)** panel, specify E next to the **Product parameters** field, and press Enter.
2. On the **Product Parameters (CCQPPRD)** panel, specify the parameter values. Required parameters are indicated by an asterisk (*).



```
CCQPPRD            Product Parameters: DB2 Admin Tool                 19:18:10
Command ===>                                                  Scroll ===> CSR

Product customization library  : TS5776.TCZ.CUST.A668.$RS22$.ADB1210
                                                                   More:     +

* General Customization

  * Customize
   *Admin Tool/OC CLIST  . 'DMTOOL.SADBCLST'                     More
   *Admin Tool DBRM  . . . 'DMTOOL.SADBDBRM'                     More
   *Admin Tool/OC EXEC . . 'DMTOOL.SADBEXEC'                     More
   *Admin Tool/OC Load . . 'DMTOOL.SADBLLIB'                     More
   *Admin Tool/OC Message  'DMTOOL.SADBMLIB'                     More
   *Admin Tool/OC Panel  . 'DMTOOL.SADBPLIB'                     More
   *Admin Tool/OC Skeleton 'DMTOOL.SADBSLIB'                     More
   *Admin Tool/OC Table  . 'DMTOOL.SADBTLIB'                     More
   *Customized Table lib . '<hlq>.CUST.SADBTLIB'
    Admin Tool hlq . . . . . . . . . . . . . DMTOOL.ADBC1PTF
.
.
.
```

Tips:

- For field-specific information, put your cursor in the input field and press F1.
- When completing any GRANT fields, always follow the principle of least privilege, which means that users are granted the minimum level of access necessary for their job functions.
- To ensure proper security on the Db2 catalog, avoid setting the following GRANT fields to PUBLIC:
  - **Default GRANT of Admin objects**
  - **GRANT ALL ON CDB tables TO field**
  - **GRANT SELECT ON xxxAUTH tables TO** (Not setting this field to PUBLIC ensures that the SYSIBM.SYSxxxAUTH tables cannot be read by PUBLIC.)
  - **GRANT SELECT ON remaining tables TO**

  Instead, specify specific authorization IDs or roles. Or, if you use RACF security or another means to control access (such as grant management), specify NONE in these fields so that no GRANT statements are generated.
- On this panel, you can specify whether you want to make DB2I and Db2 Object Comparison Tool available from the Db2 Admin Tool main menu. For instructions, see steps “2” on page 103 and “3” on page 103 in “Making DB2I and IBM Db2 Object Comparison Tool for z/OS available from the Db2 Administration Tool main menu” on page 102.
- For the **Customized Table lib** field, consider specifying an output data set name other than the shipped *hlq*.SADBTLIB. For example, specify a pre-allocated generic data set name (such as *hlq*.CUST.SADBTLIB) to store the ISPF customization table member ADBTPARM. This specification has the following advantages:
  - When upgrading Db2 Admin Tool, you do not need to run the Discover EXEC or manually move the ISPF customization table member ADBTPARM from *hlq*.SADBTLIB if the name of this shipped data set changes.
  - **Description** and **Group** information from all Db2 subsystems that are managed by TCz are properly displayed on the **Active Db2 Systems (ADB2SYS)** panel by using Launchpad mode when a common or general library (for example, *hlq*.CUST.SADBTLIB) is used to store ADBTPARM, even across LPARs when the common library is on a shared volume.

  When using this method, make sure *hlq*.CUST.SADBTLIB is also included in the concatenation when invoking Db2 Admin Tool.

  When TCz generates multiple ADBCUST jobs, a product parameters section is included in each job. Therefore, the value of the product parameter **Customized Table lib** must be the same in each ADBCUST job.

3. Press End to save your changes and exit, or issue the SAVE command to save your changes and stay on the **Product Parameters (CCQPPRD)** panel.

Related tasks

“Defining Db2 parameters” on page 106
In TCz, *Db2 parameters* are parameters for a Db2 entry.

“Defining LPAR parameters” on page 104
In TCz, *LPAR parameters* are parameters on the local LPAR that are required to customize Db2 Admin Tool.

Related information

Browsing parameters (IBM Tools Customizer for z/OS 1.1)

Defining product or component parameters (IBM Tools Customizer for z/OS 1.1)

Making DB2I and IBM Db2 Object Comparison Tool for z/OS available from the Db2 Administration Tool main menu

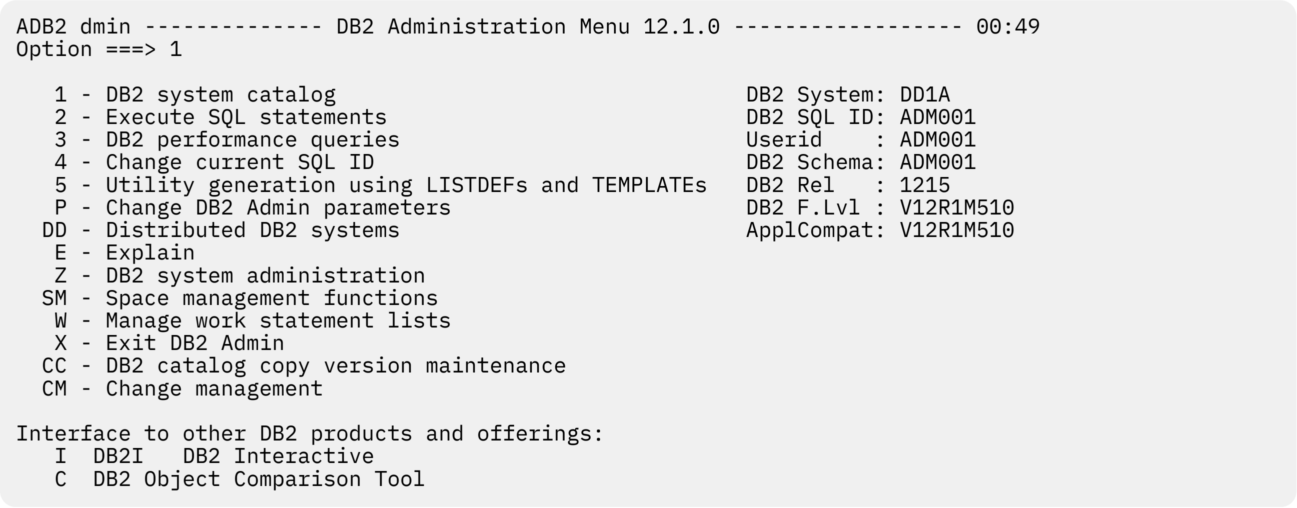
Db2 Interactive (DB2I) is a Db2 facility that enables you to perform most Db2 tasks interactively. Object Comparison Tool is a Db2 Admin Tool extension that lets you compare source and target objects. It generate reports that show the differences between the objects and jobs to apply changes to the target objects.

Before you begin

- All of the product customization steps that must be done before starting Tools Customizer are complete.
- The LPAR ISPF libraries that are required to submit the jobs are known.
- Tools Customizer is started.
- The Tools Customizer settings have been reviewed or modified, and saved.
- If you ran the Db2 Admin Tool Discover EXEC, you reviewed the values that were discovered.

About this task

When you customize Db2 Admin Tool for the first time or recustomize it, you can add DB2I and Object Comparison Tool as options in the **DB2 Administration Menu (ADB2)** panel as shown in the following figure under the section **Interface to other Db2 products and offerings**.



```
ADB2 dmin -------------- DB2 Administration Menu 12.1.0 ------------------ 00:49
Option ===> 1

   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   : 1215
   P - Change DB2 Admin parameters                     DB2 F.Lvl : V12R1M510
  DD - Distributed DB2 systems                         ApplCompat: V12R1M510
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

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

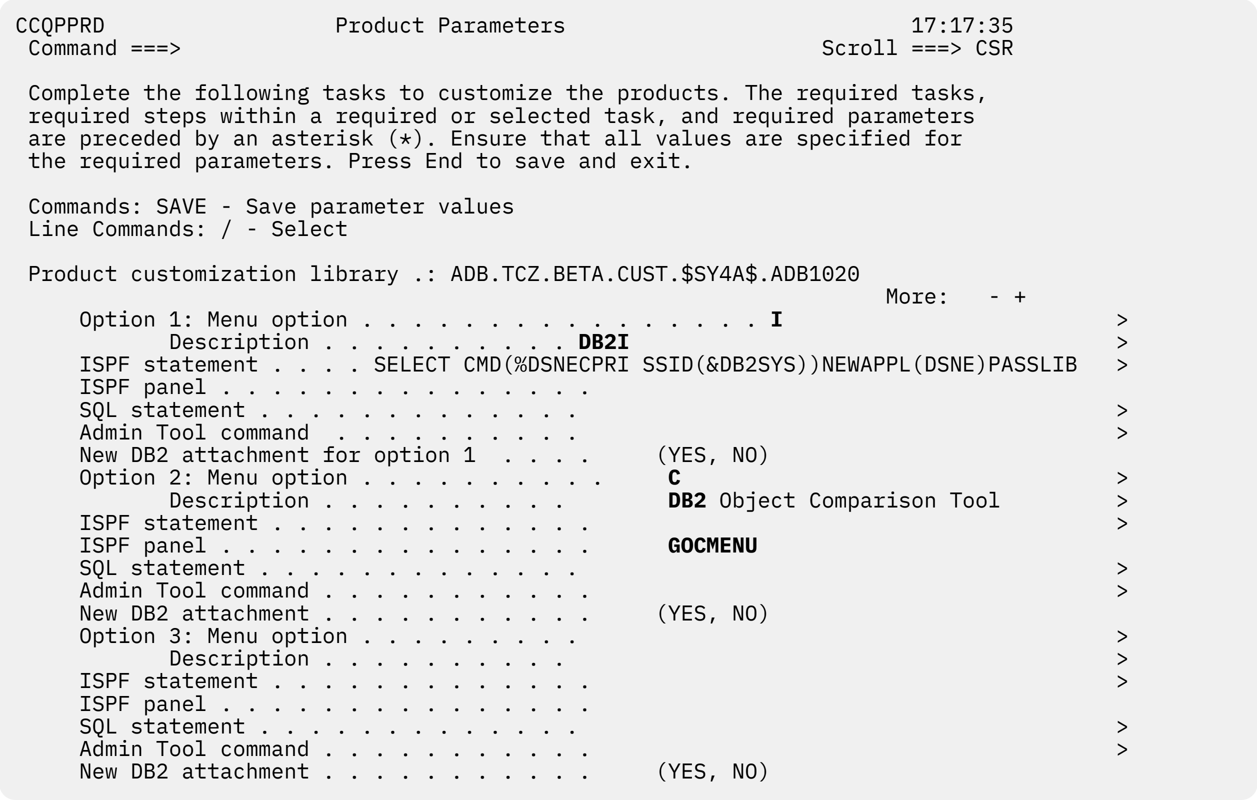
Figure 11. ***DB2 Administration Menu (ADB2)*** *panel*

Procedure

To make DB2I and IBM Db2 Object Comparison Tool for z/OS available from the Db2 Administration Tool:

1. On the **Customizer Workplace (CCQPWRK)** panel, specify E next to the **Product parameters** field, and press Enter.

The **Product Parameters (CCQPPRD)** panel is displayed as shown in the following figure:



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

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

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

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

Figure 12. Product Parameters panel

2. Complete the following **Option *n*** fields for DB2I (where *n* is a number between 1 and 10):

   Note: You might have to scroll through several pages to find these options.

   Option *n* : Menu option
   The value that is to be displayed on the **DB2 Administration Menu (ADB2)** panel. Pick a value that users are likely to associate with invoking DB2I, such as `I`.

   Description
   A description to be displayed on the **DB2 Administration Menu (ADB2)** panel next to the new menu option. For example, `DB2I`.

   ISPF statement
   The ISPF statement that Db2 Admin Tool executes for this menu option. Specify the following statement:

   ```
   SELECT CMD(%DSNECPRI SSID(&DB2SYS)) NEWAPPL(DSNE) PASSLIB
   ```

3. Complete the following **Option *n*** fields for Object Comparison Tool:

   Option *n* : Menu option
   Specify `C` or another logical value.

   Description
   Specify a description of the option, such as `Db2 Object Comparison Tool`.

   ISPF panel
   The name of the ISPF panel that Db2 Admin Tool displays for this menu option. Specify `GOCMENU`.

4. Generate the customization jobs for the Db2 subsystems (SSIDs) on which you want to have DB2I and Object Comparison Tool.
5. Submit the ADBCUST job for each of the Db2 subsystems to which you applied a customization job.

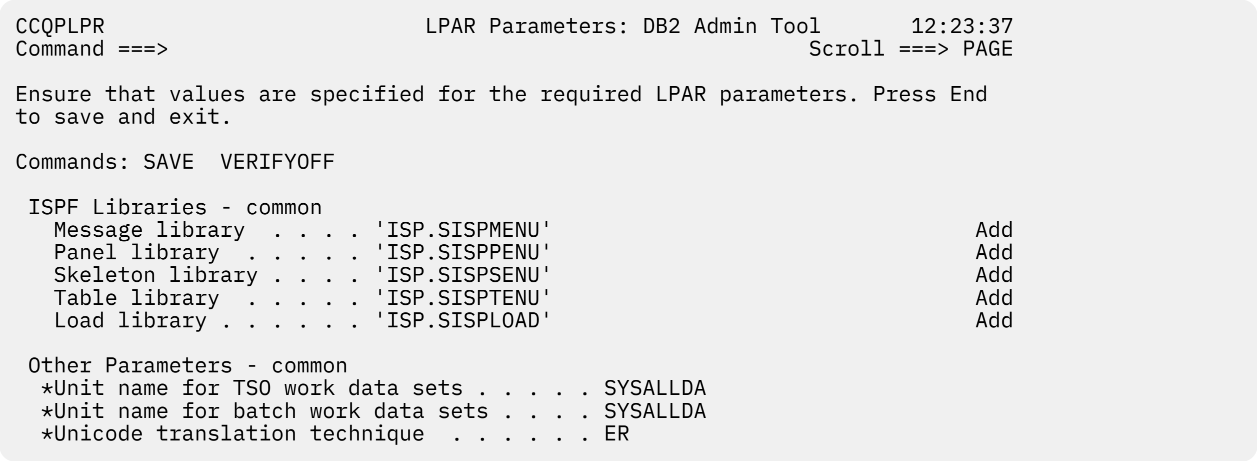
Defining LPAR parameters

In TCz, *LPAR parameters* are parameters on the local LPAR that are required to customize Db2 Admin Tool.

Procedure

To define LPAR parameters:

1. On the **Customizer Workplace (CCQPWRK)** panel, specify E next to the **LPAR parameters** field, and press Enter.
2. On the **LPAR Parameters (CCQPLPR)** panel, specify the required parameter values:



```
CCQPLPR                         LPAR Parameters: DB2 Admin Tool       12:23:37
Command ===>                                                  Scroll ===> PAGE

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

Commands: SAVE  VERIFYOFF

 ISPF Libraries - common
   Message library  . . . . 'ISP.SISPMENU'                                 Add
   Panel library  . . . . . 'ISP.SISPPENU'                                 Add
   Skeleton library . . . . 'ISP.SISPSENU'                                 Add
   Table library  . . . . . 'ISP.SISPTENU'                                 Add
   Load library . . . . . . 'ISP.SISPLOAD'                                 Add

 Other Parameters - common
  *Unit name for TSO work data sets . . . . . SYSALLDA
  *Unit name for batch work data sets . . . . SYSALLDA
  *Unicode translation technique  . . . . . . ER
```

*Figure 13. **LPAR Parameters (CCQPLPR)** panel*

Tips:

- For field-specific information, put your cursor in the input field and press F1.
- Multiple values can be specified for the libraries.
- For help with the **Unicode translation technique** value, see "Specifying a Unicode translation technique parameter value" on page 104.

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

Related tasks

"Defining Db2 Admin Tool parameters" on page 100
In TCz, *Db2 Admin Tool parameters* contain values that are specific to Db2 Admin Tool. If you ran the Db2 Admin Tool Discover EXEC, these parameter values were discovered for you; you need only to review them.

"Defining Db2 parameters" on page 106
In TCz, *Db2 parameters* are parameters for a Db2 entry.

Related information

Browsing parameters (IBM Tools Customizer for z/OS 1.1)

Defining LPAR parameters (IBM Tools Customizer for z/OS 1.1)

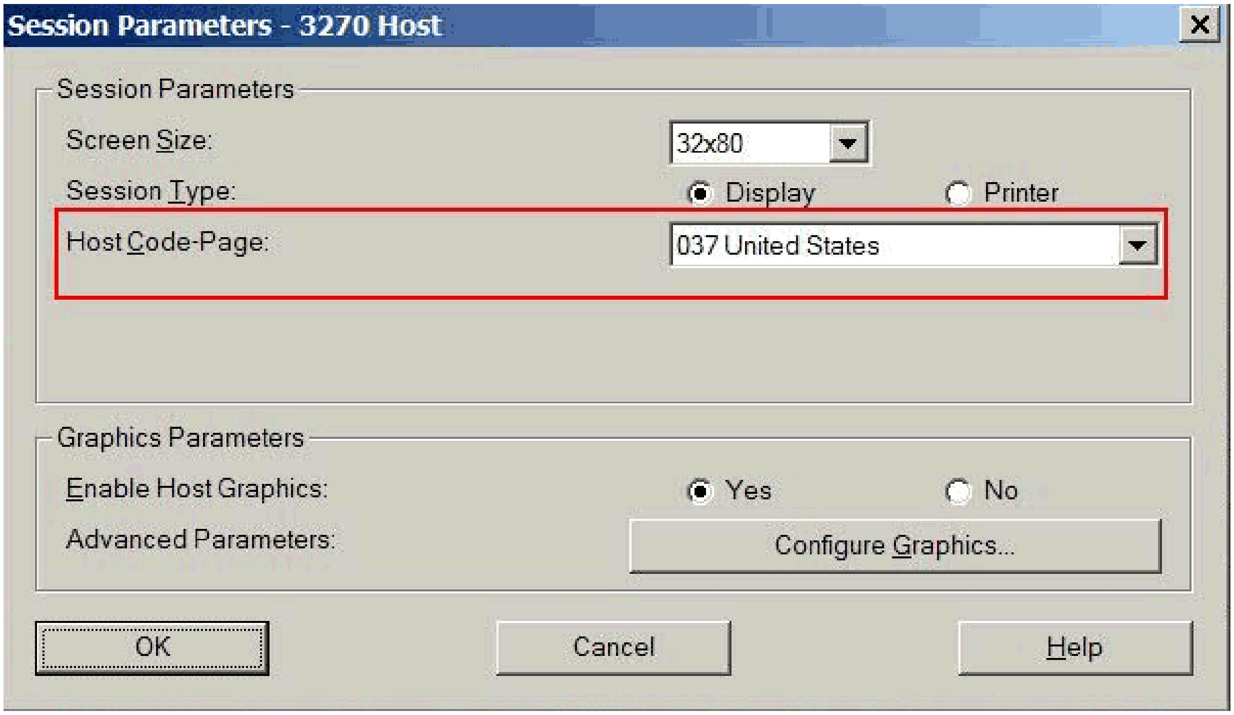
Specifying a Unicode translation technique parameter value

You might need to change the value of the **Unicode translation technique** field on the **LPAR Parameters (CCQPLPR)** panel.

The value in the field is derived from the CCSID conversion string, 01208. CCSID 01208 specifies Unicode UTF-8 data

To find the value that you need to specify:

1. Open a 3270 emulation session and find the 3270 emulation CCSID value, *xxx*, on the **Session Parameters - 3270 Host** panel, in the **HostCode-Page** field. In this example, the CCSID value is 037.



Session Parameters - 3270 Host

Session Parameters

Screen Size: 32x80

Session Type: ● Display ○ Printer

Host Code-Page: 037 United States

Graphics Parameters

Enable Host Graphics: ● Yes ○ No

Advanced Parameters: Configure Graphics...

Figure 14. Session Parameters - 3270 Host panel

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

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

Figure 15. /D UNI,ALL output

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

Defining Db2 parameters

In TCz, *Db2 parameters* are parameters for a Db2 entry.

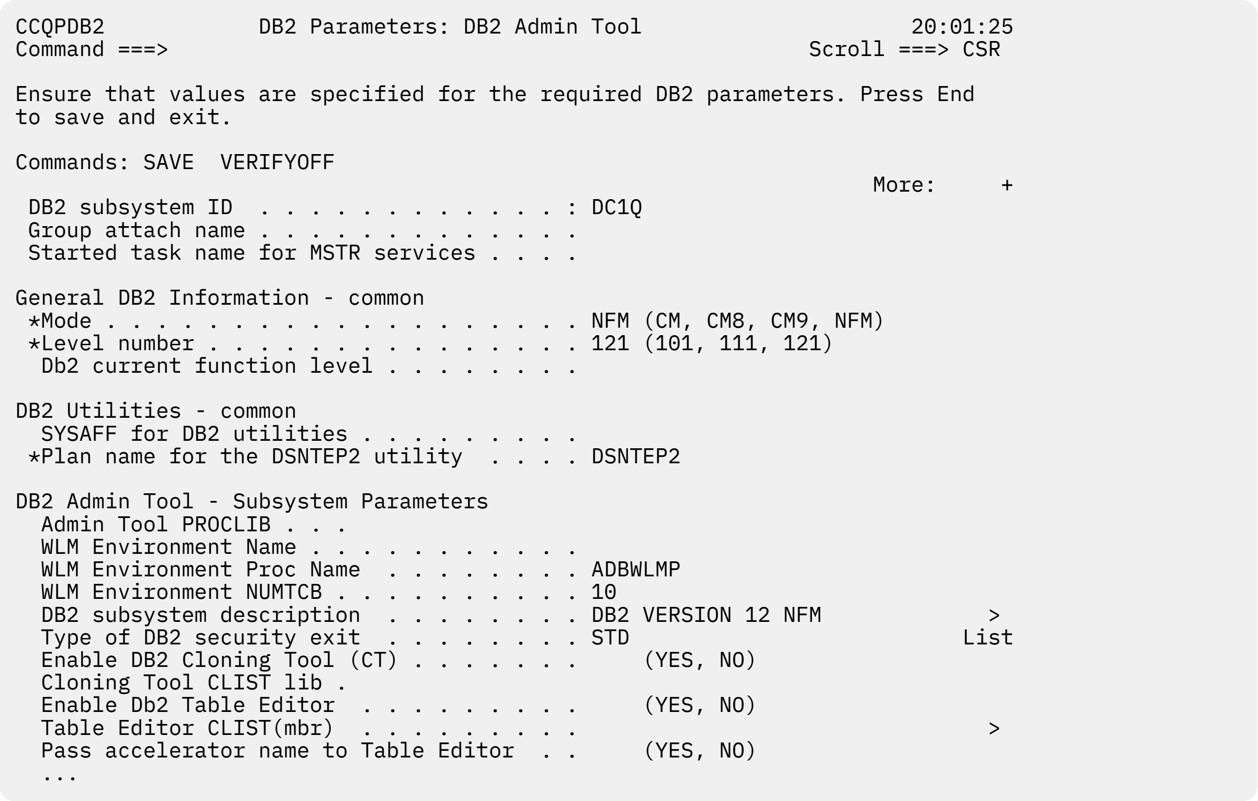
Before you begin

If you did not run the Db2 Admin Tool Discover EXEC, you must create and associate one or more Db2 entries before you can define the Db2 parameters. For instructions, see Creating and associating DB2 entries (IBM Tools Customizer for z/OS 1.1).

Procedure

To define Db2 parameters:

1. On the **Customizer Workplace (CCQPWRK)** panel, in the **Associated DB2 Entries and Parameter Status** section, specify E next to one or more Db2 entries, and press Enter.
2. On the **DB2 Parameters (CCQPDB2)** panel, specify the parameter values. Required parameters are indicated by an asterisk (*).



```
CCQPDB2            DB2 Parameters: DB2 Admin Tool                     20:01:25
Command ===>                                                  Scroll ===> CSR

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

Commands: SAVE  VERIFYOFF
                                                                   More:     +
 DB2 subsystem ID  . . . . . . . . . . . . . : DC1Q
 Group attach name . . . . . . . . . . . . . .
 Started task name for MSTR services . . . .

General DB2 Information - common
 *Mode . . . . . . . . . . . . . . . . . . . . NFM (CM, CM8, CM9, NFM)
 *Level number . . . . . . . . . . . . . . . . 121 (101, 111, 121)
  Db2 current function level . . . . . . . .

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

DB2 Admin Tool - Subsystem Parameters
  Admin Tool PROCLIB . . .
  WLM Environment Name . . . . . . . . . . .
  WLM Environment Proc Name  . . . . . . . . ADBWLMP
  WLM Environment NUMTCB . . . . . . . . . . 10
  DB2 subsystem description  . . . . . . . . DB2 VERSION 12 NFM             >
  Type of DB2 security exit  . . . . . . . . STD                          List
  Enable DB2 Cloning Tool (CT) . . . . . . .     (YES, NO)
  Cloning Tool CLIST lib .
  Enable Db2 Table Editor  . . . . . . . . .     (YES, NO)
  Table Editor CLIST(mbr)  . . . . . . . . .                                >
  Pass accelerator name to Table Editor  . .     (YES, NO)
  ...
```

Figure 16. ***DB2 Parameters (CCQPDB2)*** *panel*

Tips:

- For field-specific information, put your cursor in the input field and press F1.
- When completing any GRANT fields, always follow the principle of least privilege, which means that users are granted the minimum level of access necessary for their job functions.
- Any fields that have ADD at the end allow multiple values.
- To ensure proper security on the Db2 catalog, avoid setting the following GRANT fields to PUBLIC:
  - **GRANT ALL ON CDB tables TO field**
  - **GRANT SELECT ON xxxAUTH tables TO** (Not setting this field to PUBLIC ensures that the SYSIBM.SYSxxxAUTH tables cannot be read by PUBLIC.)
  - **GRANT SELECT ON remaining tables TO**

Instead, specify specific authorization IDs or roles. Or, if you use RACF security or another means to control access (such as grant management), specify NONE in these fields so that no GRANT statements are generated.

- Use the fields under **Db2 Admin Tool main menu - option parameters** to specify additional options to display on the main **DB2 Administration Menu (ADB2)** panel. You can specify up to 10 additional options.
- Some of the parameters on this panel are identical to parameters on the **Product Parameters (CCQPPRD)** panel. If you leave these parameters blank, Tools Customizer uses the values specified on the CCQPPRD panel. If you use unique values for specific Db2 entries, specify these values on the CCQPDB2 panel.

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

Related tasks

“Defining Db2 Admin Tool parameters” on page 100
In TCz, *Db2 Admin Tool parameters* contain values that are specific to Db2 Admin Tool. If you ran the Db2 Admin Tool Discover EXEC, these parameter values were discovered for you; you need only to review them.

“Defining LPAR parameters” on page 104
In TCz, *LPAR parameters* are parameters on the local LPAR that are required to customize Db2 Admin Tool.

Related information

Browsing parameters (IBM Tools Customizer for z/OS 1.1)

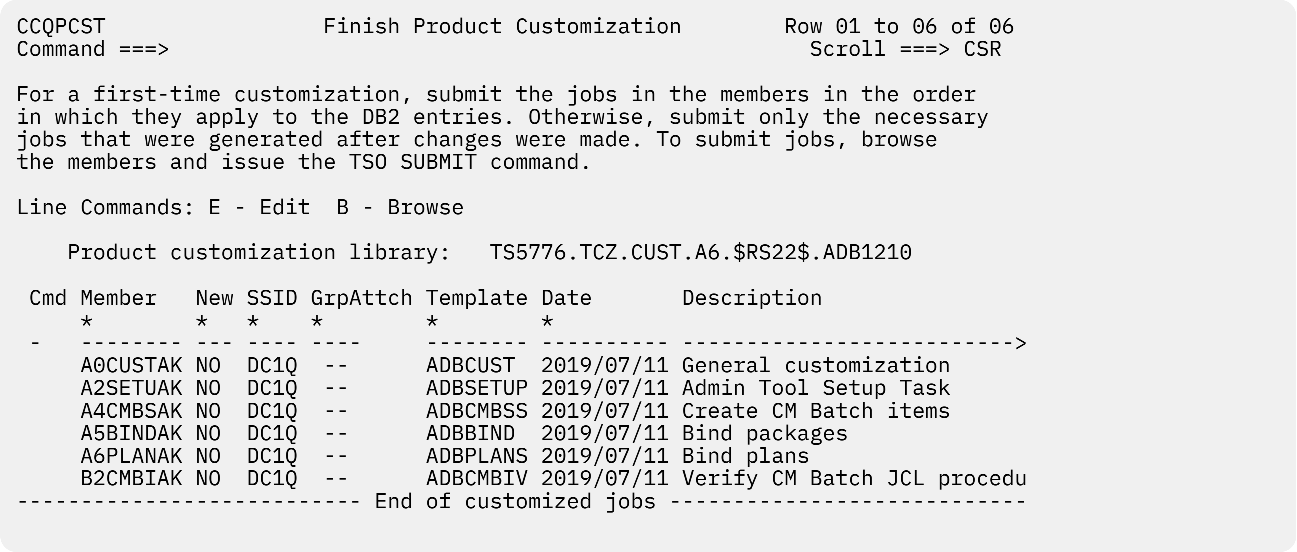
Defining Db2 parameters (IBM Tools Customizer for z/OS 1.1)

Submitting the customization jobs

After TCz generates the customization jobs for Db2 Admin Tool, you must submit them to complete the customization process.

About this task

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 01 to 06 of 06
Command ===>                                                  Scroll ===> CSR

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:   TS5776.TCZ.CUST.A6.$RS22$.ADB1210

 Cmd Member   New SSID GrpAttch Template Date       Description
     *        *   *    *        *        *
 -   -------- --- ---- ----     -------- ---------- --------------------------->
     A0CUSTAK NO  DC1Q  --      ADBCUST  2019/07/11 General customization
     A2SETUAK NO  DC1Q  --      ADBSETUP 2019/07/11 Admin Tool Setup Task
     A4CMBSAK NO  DC1Q  --      ADBCMBSS 2019/07/11 Create CM Batch items
     A5BINDAK NO  DC1Q  --      ADBBIND  2019/07/11 Bind packages
     A6PLANAK NO  DC1Q  --      ADBPLANS 2019/07/11 Bind plans
     B2CMBIAK NO  DC1Q  --      ADBCMBIV 2019/07/11 Verify CM Batch JCL procedu
--------------------------- End of customized jobs -----------------------------
```

*Figure 17. **Finish Product Customization (CCQPCST)** panel*

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

Customization jobs for Db2 entries

The members use the following naming convention:

```
<job_sequence_number><job_ID><DB2_entry_ID>
```

where

job_sequence_number

Two alphanumeric characters, A0 - Z9, that 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. Db2 Admin 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 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. Db2 Admin 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

To submit the customization jobs:

1. Submit the generated customization jobs by following your organization's process or by using the following method:

   a) Specify B or E against a customization job or the product customization library, and press Enter.

      An ISPF browsing or editing session is started.

b) Browse the customization job or each member in the library to ensure that the information is correct.

c) Run the TSO `SUBMIT` command.

2. Press End.

Results

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

Related information

Generating customization jobs (IBM Tools Customizer for z/OS 1.1)
Displaying customization jobs (IBM Tools Customizer for z/OS 1.1)
Maintaining customization jobs (IBM Tools Customizer for z/OS 1.1)

Generated customization jobs based on product parameters

Based on the parameters that you specify on the **Product Parameters (CCQPPRD)** panel, the following customization jobs are generated. SYSADM or equivalent authority is required to run the generated jobs.

Table 8. Customization jobs generated for product parameters

Parameters	Jobs generated
General Customization Customizes the general Db2 Admin Tool parameters.	ADBCUS*ab* where *ab* are alphanumeric characters that are assigned by Tools Customizer. This job is based on the ADBCUST template and is in member *job_sequence_number*CUST*DB2_entry_ID*.
Optional: Copy Fixed-Blocked (FB) CLIST/EXEC libraries to Variable-Blocked (VB) Creates VB versions of the CLIST and EXEC libraries. The data set names of the new VB libraries are the same as the FB libraries but are suffixed with ".VB".	ADBFB2VB This job is based on the ADBFB2VB template and is in member *job_sequence_number*FB2VB.

Table 8. Customization jobs generated for product parameters (continued)	
Parameters	**Jobs generated**
Admin Tool Setup Task (Create and Upgrade) Creates and upgrades objects that are used by Db2 Admin Tool.	ADBSETUP This job processes the following parameters and the related sub-parameters: • Change Management database • Checkpoint database • Catalog Copy database • Profiles History database • Reverse Engineering objects • Stored Procedure ADB2RCP • VIEW RUNSTATS objects • Indexes • GRANT on Db2 Catalog Tables **Tip:** If you want to change any of the information in ADBSETUP, go back to the **DB2 Parameters (CCQPDB2)** panel and specify different values for the relevant parameters. For example, to suppress all GRANT statements in ADBSETUP, set the **Default GRANT of Admin objects** field on this panel to NONE. Then, regenerate the jobs.
BIND: parameters Binds plans and packages.	ADBBIN*ab* where *ab* are alphanumeric characters that are assigned by Tools Customizer. This job is based on the ADBBIND template and is in member *job_sequence_number*BIND*DB2_entry_ID*.
Optional: Create sample ADBTEP2 job Tests ADBTEP2	ADBTEP*ab* where *ab* are alphanumeric characters that are assigned by Tools Customizer. This job is based on the ADBTEP2R template and is in member *job_sequence_number*TEP2*DB2_entry_ID*.
Optional: Create sample CM Batch job Tests Change Management (CM) batch.	ADBCMB*ab* where *ab* are alphanumeric characters that are assigned by Tools Customizer. This job is based on the ADBCMBIV template and is in the member *job_sequence_number*CMBI*DB2_entry_ID*

Updating the APF authorization table

The ADB2ATH and ADB2UTIL programs and TSO commands must be APF authorized to use them within Db2 Admin Tool. If the SADBLINK library, which contains ADB2ATH and ADB2UTIL, is not APF authorized, you must copy these modules to an APF-authorized library.

About this task

The ADB2ATH program is used when the Db2 security exit type (:secexit.) is specified as AUTH. The ADB2UTIL program is used when the ADBTEP2 program runs Db2 utilities.

Procedure

To update the APF authorization table:

1. Copy ADB2ATH and ADB2UTIL from *high-level*.SADBLINK to an APF-authorized library.

   If the APF-authorized library is not in the system link list, it must be registered as the **Admin Tool APF Library** on the **Product Parameters (CCQPPRD)** panel.

   Using two different versions of Db2 Admin Tool on the same Db2 subsystem: If you are installing and running two or more different releases of Db2 Admin Tool on the same Db2 subsystem, ensure that you use the ADB2ATH and ADB2UTIL authorized programs from the product tape for the higher release when you copy them to the APF-authorized library in your system link list.

   If you plan to use Db2 High Performance Unload (HPU) in a work statement list (WSL), also authorize the HPU main program, INZUTILB by ensuring it is in an APF-authorized library.
2. Modify SYS1.PARMLIB(IKJTSO*xx*) and add programs ADB2ATH and ADB2UTIL to both the AUTHPGM and AUTHTSF sections, as shown in the following figure:

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

   Figure 18. Adding programs ADB2ATH and ADB2UTIL to IKJTSOxx

   This action ensures that the TSO service facility invokes ADB2ATH and ADB2UTIL as authorized programs.

   If plan to use HPU in a WSL, also add INZUTILB to the AUTHPGM and AUTHTSF sections.
3. Activate the changes at the next IPL or immediately by issuing the following TSO/E command:

```
PARMLIB UPDATE(xx)
```

Verifying activation of limited functionality

After submitting the BIND job (and the fixed-to-variable-block conversion job, if necessary), you can use Db2 Admin Tool with limited functionality. At this point, you might want to verify that Db2 Admin Tool is operational.

After you ensure that Db2 Admin Tool is operational (by completing the following procedure), you can enable full use of it by ensuring that the Db2 libraries are allocated to your TSO session.

Procedure

To verifying activation of limited functionality:

1. Determine where the Db2 DSN command is installed.
2. Make the Db2 Admin Tool ISPF and TSO libraries available to your ISPF session by completing one of the following actions:

   a. Add the Db2 Admin Tool product libraries to your TSO LOGON PROC. The following table shows the libraries that you can allocate to your TSO LOGON PROC.

Table 9. Libraries to allocate to your TSO LOGON PROC

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

b. Write a small CLIST that runs the ADBL CLIST.

3. Start Db2 Admin Tool according to the option that you chose in “2” on page 111.

- If you chose option a, issue the following command:

  ```
  TSO %ADBL
  ```

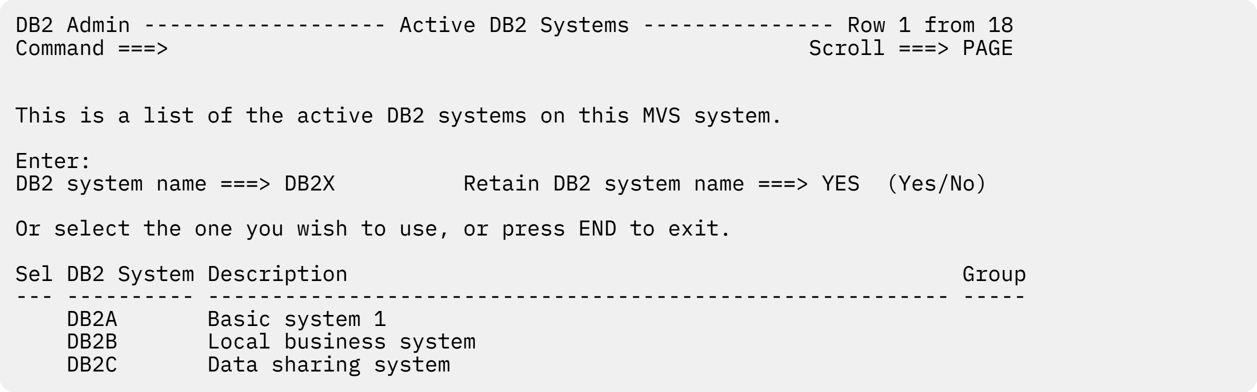
  If the Db2 DSN command is not in the linklist, you need to specify the data set name of the Db2 load module library where the DSN command resides. Specify this value as a parameter when you issue the command as follows:

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

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

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

If more than one Db2 subsystem is active, the **Active DB2 Systems (ADB2SYS)** panel is displayed. This panel lists all Db2 subsystems, as shown in the following figure:

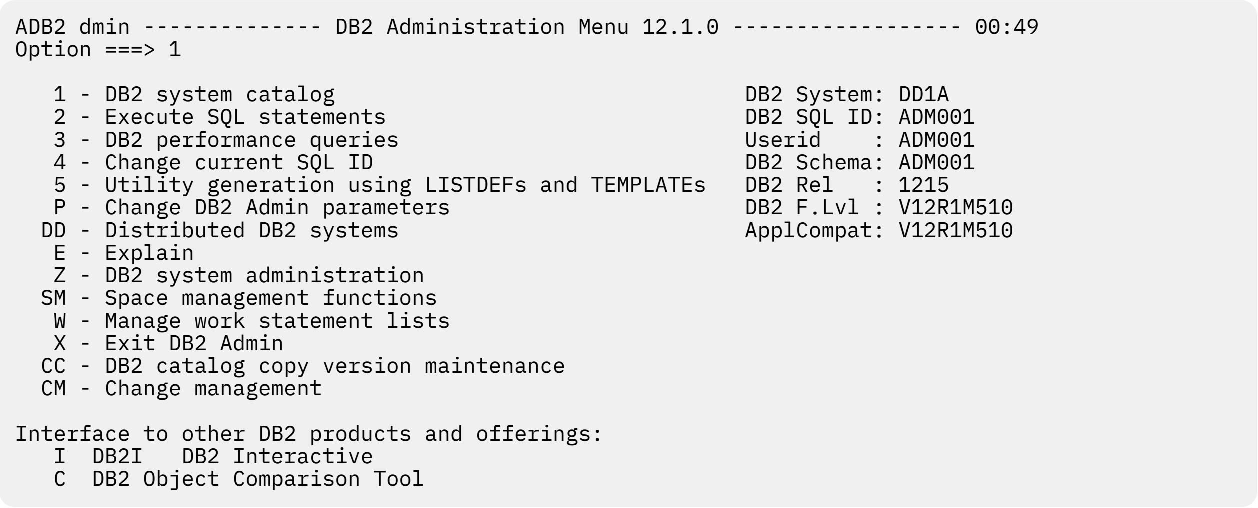


```
DB2 Admin ------------------- Active DB2 Systems --------------- Row 1 from 18
Command ===>                                                  Scroll ===> PAGE


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

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

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

Sel DB2 System Description                                                Group
--- ---------- ---------------------------------------------------------- -----
    DB2A       Basic system 1
    DB2B       Local business system
    DB2C       Data sharing system
```

*Figure 19. **Active DB2 Systems (ADB2SYS)** panel*

4. Select the Db2 subsystem that you want to use, and press Enter.

   The **DB2 Administration Menu (ADB2)** panel is displayed, as shown in the following figure:



```
ADB2 dmin -------------- DB2 Administration Menu 12.1.0 ------------------ 00:49
Option ===> 1

   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   : 1215
   P - Change DB2 Admin parameters                      DB2 F.Lvl : V12R1M510
  DD - Distributed DB2 systems                          ApplCompat: V12R1M510
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

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

Figure 20. ***DB2 Administration Menu (ADB2)*** *panel*

Attention: If an SQL error occurs, ensure that the application plan (ADB) and the packages (ADBMAIN, ADB2GET, and ADB2CON) are bound correctly on the Db2 subsystem that you are using. Also, verify that you have granted EXECUTE on the application plan ADB to the user IDs that are running Db2 Admin Tool. Then, go back to step "2" on page 111.

5. Verify that Db2 Admin Tool is available with limited functionality by completing the following steps:

   a) Specify option **1**. to go to the Db2 System Catalog.

   b) On the **System Catalog (ADB21)** panel, specify option **T**.

   c) On the **Tables, Views, and Aliases (ADB21T)** panel panel, specify SYSDUMMY1 for the name and SYSIBM for the owner, and press Enter.

   You should see a row that describes the table SYSDUMMY1 owned by user SYSIBM. This result verifies that Db2 Admin Tool can retrieve rows from the Db2 catalog.

   d) Issue the DDL line command next to the SYSDUMMY1 row.

   You should see a CREATE TABLE statement that can be used to create the table.

Example

To invoke Db2 Admin Tool for Db2 subsystem ABCD, enter the following command on the command line on the ISPF main menu:

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

To invoke the **Launchpad (ADBDMT)** panel panel, which enables you to continue using Db2 Admin Tool functions, or to invoke another Db2 tool, enter the following command on the command line on the ISPF command processor panel (usually ISPF option 6):

```
%ADBL DMT
```

To directly invoke Db2 Admin Tool and display all of the active Db2 subsystems that are available to you, enter the following command on the command line on the ISPF command processor panel (usually ISPF option 6):

```
%ADBL SHOW
```

What to do next

To enable the full use of Db2 Admin Tool, check your TSO LOGON PROC and your link list definition to verify that the Db2 libraries are available to your TSO session.

Specifying naming conventions

You can set up Db2 Admin Tool to use your local naming conventions for items such as data sets, utility IDs (UIDs), and plan names. The default settings are provided in skeleton ADB2UCUU. Update skeleton member SADBSLIB(ADB2UCUS) to override any of the variable default settings in SADBSLIB(ADB2UCUU) to site-specific standards. Do not update SADBSLIB(ADB2UCUU).

Review SADBSLIB(ADB2UCUU) and use skeleton member SADBSLIB(ADB2UCUS) to override any variable default settings to site-specific standards. Over 250 variable settings are provided. Each variable setting is documented in member ADB2UCUU. For example, you can set the default plan name for sample programs DSNTIAD and DSNTIAUL.

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

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

The following list includes some of the variables that you can use in the specified naming conventions. For a complete list of variables, see skeleton member ADB2UCUT.

Variable:
Description:

&AJDATE
Julian date (YYDDD)

&AJDAY
Julian day (DDD)

&AYEAR4
4-digit year (YYYY)

&AGDATE
Gregorian date (YYMMDD)

&ANMON
Numeric month (MM)

&ADAY
Day (DD)

&AYEAR
2-digit year (YY)

&ACMON
3-character month (XXX)

&ATIME
Time (HHMMSS)

&ATIME7
Time with tenths of seconds (HHMMSST)

&ATIME4
Time without seconds (HHMM)

&AHOUR
Hour (HH)

&AMIN
Minute (MM)

&ASEC
Seconds (SS)

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

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

The variable ASYCPY1 is shipped as:

```
)SET ASYCPY1 = &PREFIX..&DB2SYS..IC.&DBNAME..&NAME.(+1)
```

To change the high-level qualifier from the current TSO PREFIX to MYHLQ, specify:

```
)SET ASYCPY1 = MYHLQ.&DB2SYS..IC.&DBNAME..&NAME.(+1) /* CHANGE HLQ TO FIXED STRING
```

To change the second-level qualifier from the Db2 subsystem ID to TEST, specify:

```
)SET ASYCPY1 = &PREFIX..TEST.IC.&DBNAME..&NAME.(+1)  /* CHANGE SUBSYSTEM TO 'TEST'
```

To insert a high-level qualifier of MYHLQ in front of the current TSO PREFIX and to remove the Db2 database name, specify:

```
)SET ASYCPY1 = MYHLQ.&PREFIX..&DB2SYS..IC.&NAME.(+1)
               /* CHANGE HLQ TO FIXED STRING,
               /* INCLUDE PREFIX, REMOVE DBNAME
```

To use sequential data sets rather than a GDG data set, specify a data set name that contains date and time values to generate unique data set names:

```
)SET ASYCPY1 = &PREFIX..IC.&DBNAME..&NAME..D&AJDATE..T&ATIME
```

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

The variables PREFXUID, LOADUID, and UNLODUID are included as:

```
)SET PREFXUID = &Z
)SET LOADUID = &PREFXUID
)SET UNLODUID = &PREFXUID
```

To change the LOAD and UNLOAD UIDs such that they contain the TSO user ID, a time stamp, and a utility type identifier, specify:

```
)SET PREFXUID = &ZUSER.&ATIME
)SET LOADUID = &PREFXUID.LD
)SET UNLODUID = &PREFXUID.UL
```

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

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

To change the LOAD and UNLOAD UIDs such that they contain the TSO user ID and a time stamp with tenths of seconds (USERID.HHMMSST), specify:

```
)SET PREFXUID = &ZUSER..&ATIME7
)SET LOADUID = &PREFXUID
)SET UNLODUID = &PREFXUID
```

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

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

Restrictions:

- When modifying data set names, ensure that data set names do not extend beyond column 71 in the ADB2UCUS data set. Any characters beyond column 71 are truncated.
- Data set names, including the periods, cannot be greater than 44 bytes in length. Ensure that generated data set names are not longer than 44 bytes.
- Utility IDs (UIDs), including the periods, cannot be greater than 16 bytes in length. Ensure that generated UIDs are not longer than 16 bytes.
- Utility ID (UID) customization does not apply to UIDs in work statement lists (WSLs).

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

After testing is complete, you can use an SMP/E USERMOD to update the Db2 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.

Granting SELECT access on catalog tables

Db2 Admin Tool uses dynamic SQL against the Db2 catalog. Therefore, if you plan to make Db2 Admin Tool available to a large number of users, you should specify those IDs that are authorized to see the catalog. IBM recommends that you do not grant this access to PUBLIC.

Recommendation: When granting access, always follow the principle of least privilege, which means that users are granted the minimum level of access necessary for their job functions.

Procedure

Generate the TCz jobs for the site-specific Db2 subsystem ID (SSID) and submit them in the order that they are listed on the TCz **Finish Product Customization (CCQPCST)** panel.

Optimizing ADMIN_INFO_SYSPARM and DSNZPARM settings for GEN and DDL

The Db2 Admin Tool Reverse Engineering (GEN) function uses the ADMIN_INFO_SYSPARM stored procedure to read Db2 subsystem parameter values from the Db2 DSNZPARM module.

About this task

If ADMIN_INFO_SYSPARM does not complete normally, the Reverse Engineering function waits until ADMIN_INFO_SYSPARM times out. In this case, the GEN or DDL commands might run longer than necessary as a result of Db2 waiting for the timeout value for stored procedures to be reached.

Procedure

To optimize the performance of the GEN and DDL commands, optimize ADMIN_INFO_SYSPARM and the DSNZPARM settings as follows:

- Verify that the ADMIN_INFO_SYSPARM stored procedure is operational.
- Verify that the STORTIME subsystem parameter (in macro DSN6SYSP) is set to a reasonable waiting time. You might want to specify a lower value for STORTIME.

Related tasks

“Generating SQL to re-create a Db2 object (reverse engineering)” on page 357
Before making changes to a Db2 object, you might find it useful to generate the SQL statements that are required to re-create that object. Generating this SQL ensures that the changes are applied to the current definition and that the original object definitions are available for fallback purposes.

Related reference

TIMEOUT VALUE field (STORTIME subsystem parameter) (Db2 12 for z/OS)

Defining the provided stored procedures

If you plan to use any of the stored procedures that are provided by Db2 Admin Tool, define those procedures during the customization process.

About this task

Db2 Admin Tool provides the following stored procedures:

Procedure name	Description
ADB2ME2	Estimates the number of extents needed
ADB2MES	Provides space estimates for table spaces
ADB2MEX	Provides space estimates for index spaces
ADB2RE	Generates SQL for objects from the Db2 catalog
ADBGDDL	Returns the DDL for a single object from the Db2 catalog

Procedure

To define the provided stored procedures:

1. In TCz, run the generated ADBSETUP job to create the stored procedures.

   This job also creates the required temporary tables and the BIND PACKAGE command for the ADB2RE and ADBGDDL stored procedures on the Db2 subsystem that is to use reverse engineering.
2. For those stored procedures written in REXX (ADB2MES, ADB2MEX, and ADB2ME2), edit the JCL startup procedure for the associated WLM environment as follows and then refresh the WLM environment.
   - Add a SYSEXEC DD statement with the library that contains these stored procedures. For example:

     ```
     //SYSEXEC DD DSN=DMTOOL.SADBEXEC,DISP=SHR
     ```
   - Ensure that `NUMTCB=1` is specified for the WLM environment.
3. If the SADBLLIB load library is not included in the JCL startup procedure for the WLM environment, copy the following load module members from SADBLLIB to one of the libraries in STEPLIB in the WLM-managed stored procedure address space:
   - ADB2RE
   - ADB2ME2
   - ADB2MES
   - ADB2MEX
   - ADBGDDL
   - ADBPMSG
   - ADB0001
   - ADB1000
   - IADBMSGT
   - UADBMSGT
   - ADBMSGT
4. Optional: To verify that the stored procedures were created correctly, call them by navigating to the **Stored Procedures (ADB21O)** panel [option `1.O` from the main **DB2 Administration Menu (ADB2)** panel] and issuing the CALL line command for each of the procedures.

   Alternatively, you can manually execute an SQL CALL statement for each of the procedures.

   For CALL syntax for each procedure, see the following information:

-
-
- “ADB2ME2 stored procedure” on page 877
- “ADB2MES stored procedure” on page 879
- “ADB2MEX stored procedure” on page 882
- “ADB2RE stored procedure” on page 370
- “ADBGDDL stored procedure” on page 886
-

Setting up the provided REST APIs

To call any of the REST service APIs that are provided by Db2 Admin Tool, you must first define the provided stored procedures and then bind them as REST services.

About this task

For a list of REST APIs provided by Db2 Admin Tool and their corresponding stored procedures, see “Provided REST APIs” on page 876.

Procedure

To set up the provided REST APIs:

1. Create the stored procedures.
2. Run one or more of the following sample jobs to bind the stored procedures as a REST service:

Sample job	Stored procedures that it binds as a REST service
ADB2MER	ADB2MES, ADB2MEX, ADB2ME2
ADBSERV	ADBGDDL

Related tasks

“Calling a provided REST API” on page 876
You can call certain features in Db2 Admin Tool as REST APIs by using one of the provided stored procedures. For example, you can get space estimates for index spaces by using the ADB2MEX stored procedure.

Related information

BIND SERVICE (DSN) (Db2 12 for z/OS documentation)

Enabling distributed support

You can use Db2 Admin Tool on remote Db2 systems. This functionality is called *distributed support*. You partially enable distributed support when you customize Db2 Admin Tool with Tools Customizer. To completely enable distributed support, complete the following procedure.

About this task

On remote systems, you can perform the following tasks through Db2 Admin Tool:

- Build utility jobs and submit them to run on remote systems
- Perform alter and migrate functions for remote systems
- Issue SQL statements against remote systems
- Issue distributed GRANT and REVOKE commands
- Issue other commands on remote systems

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

Procedure

Copy the appropriate load module to the load data set.

Option	Description
To enable distributed support:	Copy the distributed load module ADB2RCP to the load data set for the default stored procedure address space on the Db2 subsystem.
To enable distributed support and registration of a multi-target change registration on a target system using DRDA access:	Copy the load module ADBCRSP to the load data set for the default stored procedure address space on the Db2 subsystem.

Improving performance when making Db2 Admin Tool available to users

You can eliminate the need for performing ISPF LIBDEFs each time that Db2 Admin Tool is invoked, which significantly reduces Db2 Admin Tool start-up time.

Procedure

When you make Db2 Admin Tool available to users, use one of the following methods to eliminate the need for ISPF LIBDEFs and thus improve performance when invoking the tool:

- Copy the Db2 Admin Tool ISPF and TSO libraries to your standard libraries. Your standard libraries are allocated in your TSO LOGON procedure or are allocated dynamically before you invoke ISPF.
- Allocate the Db2 Admin Tool target libraries in the TSO LOGON procedure or dynamically before you invoke ISPF.

  Tip: If possible, define the libraries that you are using for Db2 Admin Tool (and all of the libraries allocated on the same DD statements before the ones that you are using for Db2 Admin Tool) to LLA with the FREEZE option. This approach significantly reduces the number of input/outputs (I/Os) and the I/O time used when ISPF and TSO perform a search for Db2 Admin Tool members in the concatenation sequence.

Tailoring Authorization Switching

Authorization Switching is a facility within Db2 Admin Tool that is used to execute DDL and DCL under the authority of another user. The facility does not cover other statement types, including Db2 utility commands and DSN subcommands such as FREE PACKAGE and BIND PLAN. This other user is called the *auth-switch ID*, and the ID that submits the job is called the *submitter*.

About this task

Deprecation notice: Authorization switching is deprecated in Db2 Admin Tool. For more information, see "Deprecated functions and functions that are no longer supported in Db2 Admin Tool 12.1" on page 64.

Alter Tablespace ALT, Alter Table ALT, WSLs, Change Management, Change Management batch, and IBM Db2 Object Comparison Tool for z/OS make use of authorization switching. These functions allow table spaces and tables to be redefined, which requires that they, and any dependent objects, be dropped and re-created. However, the job submitter might not have the necessary authority to rebuild all the objects and authorizations. Authorization switching allows the job submitter to use an ID that does have the authority to run the DDL to rebuild the objects.

Before Db2 Admin Tool Authorization Switching can be used, some additional installation steps must be performed to enable and protect it.

To complete the installation of Db2 Admin Tool Authorization Switching:

Procedure

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

Tip: When Db2 Admin Tool Authorization Switching is enabled for a Db2 subsystem, create a RACF profile to protect the facility from unauthorized use. When DDL that is enabled with Db2 Admin Tool Authorization Switching capability is run, a RACF access check is made to a resource that is intended to protect the use of a given Authorization Switching ID on the Db2 subsystem. The resource is within the IBM-supplied RACF FACILITY class in the following form:

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

Example

If the Db2 subsystem is DSN and the desired authorization ID to use is SYSADMZ1, the RACF resource name that Db2 Admin Tool generates is:

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

For Db2 Admin Tool Authorization Switching to proceed, the job submitter requires READ authority to the profile that protects this resource. The standard RACF profile rules apply for this resource. An installation can use general or more granular profile controls, as listed in the following table.

Table 10. Controlling the granularity of profiles.

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

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

Restriction: Db2 Admin Tool Authorization Switching requires that the RRS Attach Facility (RRSAF) of Db2 for z/OS is available.

Invoking Db2 Admin Tool

Use one of the provided sample execs to invoke Db2 Admin Tool. Do not modify the ADBL CLIST.

About this task

The ADBL CLIST in the SADBCLST library invokes the Db2 Admin Tool main menu. The sample execs in the procedure below execute this CLIST with your own parameters.

Procedure

To invoke Db2 Admin Tool, use one of the following sample execs:

- **Method 1:**

```
/*Rexx*/
Arg Option

Parms = "SYSTEM(????) PLAN(ADB)",
        "LIST(''DMTOOL.ADBC1PTF DMTOOL.GOCC1PTF'')",
        "LISTPRE(''SADB SGOC'') PROD(.)",
        Option

"EXEC 'DMTOOL.SADBCLST(ADBL)'" "'"Parms"'"
```

Note: The SYSTEM() parameter is optional.

The following example of Method 1 concatenates a user set of libraries (for example, ADB.DEVCUST.ISP*) ahead of the Db2 Admin Tool product libraries:

```
/*Rexx*/
Arg Option

Parms = "SYSTEM(????) PLAN(ADB)",
        "LIST(''ADB.DEVCUST DMTOOL.ADBC1PTF DMTOOL.GOCC1PTF'')",
        "LISTPRE(''ISP SADB SGOC'') PROD(.)",
        Option

"EXEC 'DMTOOL.SADBCLSTDBCLST(ADBL)'" "'"Parms"'"
```

- **Method 2:** This method uses the PROD, PRODADD, and USERADD parameters.

```
/*Rexx*/
Arg Option

Parms = "SYSTEM(????) PLAN(ADB)",
        "USERADD(''ADB.DEVCUST'')     USERPRE(ISP)",
        "PRODADD(''DMTOOL.GOCC1PTF'') LIBAPRE(SGOC)",
        "PROD(''DMTOOL.ADBC1PTF'')    LIBPRE(SADB)",
        Option

"EXEC 'DMTOOL.SADBCLST(ADBL)'" "'"Parms"'"
```

Both methods result in the following concatenations:

Library	Concatenation
Load library	ADB.DEVCUST.ISPLLIB DMTOOL.GOCC1PTF.SGOCLLIB DMTOOL.ADBC1PTF.SADBLLIB
Message library	ADB.DEVCUST.ISPMLIB DMTOOL.GOCC1PTF.SGOCMLIB DMTOOL.ADBC1PTF.SADBMLIB
Panel library	ADB.DEVCUST.ISPPLIB DMTOOL.GOCC1PTF.SGOCPLIB DMTOOL.ADBC1PTF.SADBPLIB
Skeleton library	ADB.DEVCUST.ISPSLIB DMTOOL.GOCC1PTF.SGOCSLIB DMTOOL.ADBC1PTF.SADBSLIB
Table library	ADB.DEVCUST.ISPTLIB DMTOOL.GOCC1PTF.SGOCTLIB DMTOOL.ADBC1PTF.SADBTLIB
CLIST library	ADB.DEVCUST.CLIST DMTOOL.GOCC1PTF.SGOCCLST DMTOOL.SADBCLST
Exec library	ADB.DEVCUST.EXEC DMTOOL.GOCC1PTF.SGOCEXEC DMTOOL.ADBC1PTF.SADBEXEC

You can customize the sample exec to use the following parameters:

ASUSER

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

Restriction:

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

CMD

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

Example:

```
CMD(''CAT T SYSTEM01%.TEST%'')
```

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

CMOWN

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

The following examples show how you can use the CMOWN parameter:

```
CMOWN(CMDBADM)
CMOWN(-)
```

DASD

The unit name for batch work data sets. If you use a minus sign with this parameter, the value in the **Unit name for batch data sets** field on the **LPAR Parameters** panel is overridden by the Db2 Admin Tool default, which is SYSDA.

The following examples show how you can use the DASD parameter:

```
DASD(SYSALLDA)
DASD(-)
```

DB2LLIB

List of the Db2 product load module libraries where Db2 is installed if Db2 is not in the linklist.

DEBUG

Use this parameter only at the request of IBM Software Support.

DMT

You can use the DMT parameter to access the **DB2 Tools Launchpad** panel. From this panel, you can either invoke a Db2 tool or you can continue to use the Db2 Admin Tool functions that are described in this information. If you do not use the DMT parameter, you go directly to the Db2 Admin Tool functions, but you cannot launch other Db2 tools from within Db2 Admin Tool.

Restriction: If there is no active ISPF LIBDEF data set for table input library ISPTLIB, the Db2 Tools Launchpad can not be accessed correctly when using the DMT parameter.

DMTID

Indicates which library from the Db2 Tools Launchpad TLIB list you want to select by default. You can specify this parameter from your local front-end panel, CLIST, or from a REXX exec that invokes the ADBL CLIST. You will specify a number in parenthesis, for example, dmtid(2). An S will be placed in the Sel field for the row that you indicate (in this example, the second row) for the library that you want to be the default. This library is displayed in the panel to show where the update will be written. If the number you enter exceeds the number of rows, an S will be placed in the last row. If Launchpad is not active, then DMTID is ignored.

DUMP

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

INSTALL

Installation name.

JES

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

LIBAPRE

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

LIBPRE

The prefix for Db2 Admin Tool libraries. This prefix designates the first set of characters (up to four) in the final qualifier of the Db2 Admin Tool libraries.

The default is SADB.

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

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

LIBDEF(YES|NO)

Specify YES to enable LIBDEF/ALTLIB usage to dynamically allocate Db2 Admin Tool product libraries. YES is the default.

Specify NO to disable LIBDEF/ALTLIB usage. Db2 Admin Tool product libraries must then be pre-allocated by your TSO logon procedure.

Note: If you specify the parameter DMT with LIBDEF(NO) to access Db2 Tools Launchpad, you must ensure that there is already an active ISPF LIBDEF data set for the table input library ISPTLIB. If there is no active ISPF LIBDEF data set, you must first perform an ISPF LIBDEF statement for the library ISPTLIB, Otherwise, you can not access the Db2 Tools Launchpad if you allocate only the Db2 Admin libraries in your TSO logon procedure.

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

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

Example:

`%LIBDEF`

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

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

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

LIST

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

LISTPARM

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

LISTPRE

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

NEWAPPL

The ISPF application ID. NEWAPPL identifies the member name in which the ISPF profile variables are saved for Db2 Admin Tool. The default value for NEWAPPL is null with an application ID of ISR.

The following example shows the recommended value for the NEWAPPL parameter:

```
NEWAPPL(ADB)
```

PANEL
The panel name for the Db2 Admin Tool panel that is displayed first. The default is ADB2.

PGM
The name of the Db2 Admin Tool main program. The default is ADBMAIN.

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

PROD
Use the PROD parameter to specify the high-level qualifier of the Db2 Admin Tool product libraries. To disable, specify PROD(.)

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

QTAB
Use this parameter, which lists open ISPF tables at the beginning and end of a Db2 Admin Tool session, only at the request of your IBM service representative.

SECEXIT
The Db2 security exit type. The possible values are STD (the default), SAMPLE, AUTH, OWN, and NOCALL. If you use a minus sign with this parameter, the value set for the :secexit. tag or the **DB2 Security exit type** field on the **Product Parameters** panel is overridden by the Db2 Admin Tool default, which is STD.

The following examples show how you can use the SECEXIT parameter:

```
SECEXIT(AUTH)
SECEXIT(-)
```

SHOW
Use the SHOW parameter to start your Db2 Admin Tool session with a panel that shows all of the active Db2 subsystems that are available to you.

SYSTEM(*ssid*)
Use the SYSTEM(*ssid*) parameter to directly access a specific Db2 subsystem. This parameter is ignored if the SHOW parameter is specified.

USER
To activate the CLIST and EXEC libraries that are allocated to the SYSUPROC and SYSUEXEC DD names, issue an ALTLIB USER statement after ALTLIB APPLICATION. These libraries are then searched before searching the Db2 Admin Tool libraries.

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

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

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

VIO
The unit name for TSO work data sets. If you use a minus sign with this parameter, the value in the **Unit name for TSO work data sets** field on the **LPAR Parameters** panel is overridden by the Db2 Admin Tool default, which is VIO.

The following examples show how you can use the VIO parameter:

```
VIO(SYSALLDA)
VIO(-)
```

Setting global variables for Db2 Admin Tool

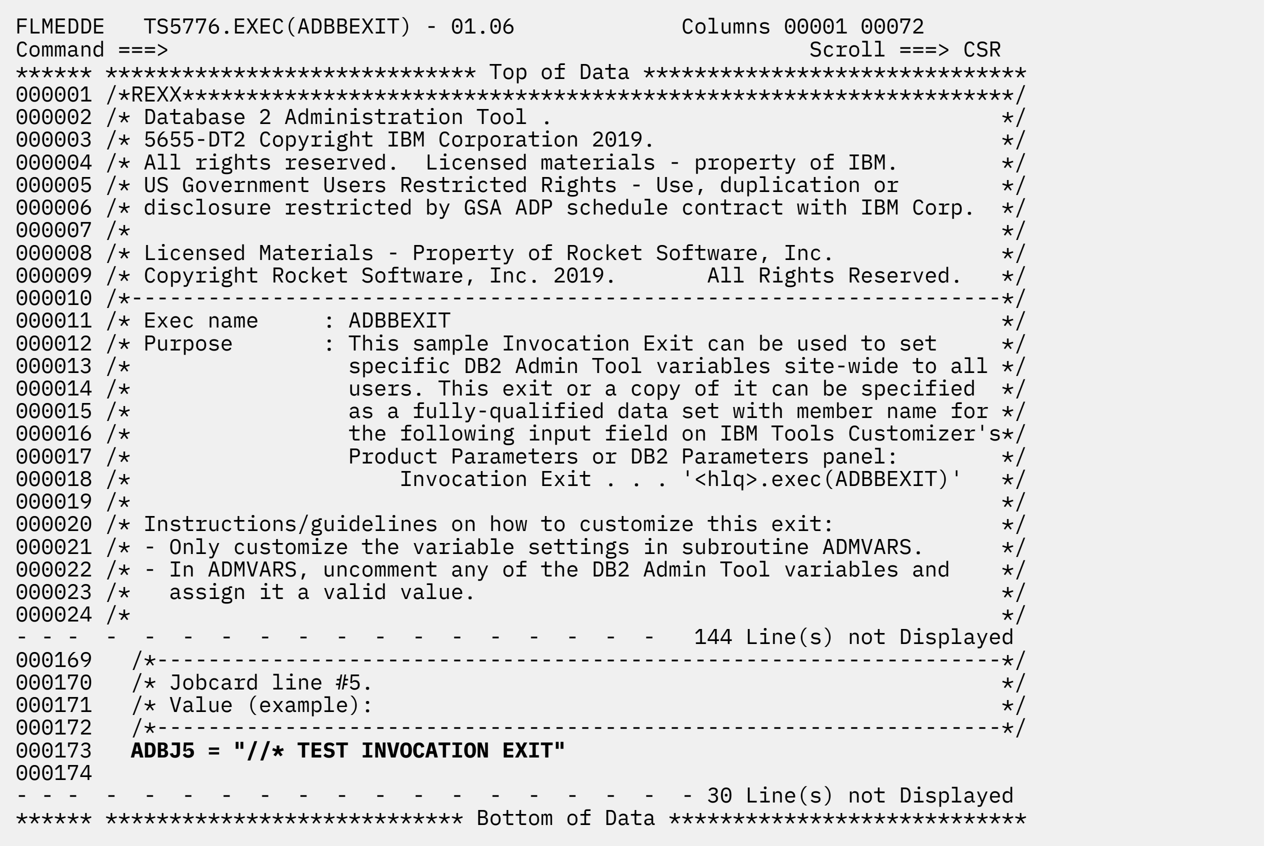
To set Db2 Administration Tool variables site-wide for all users, use an invocation exit. An *invocation exit* is a REXX exec that automatically executes before Db2 Admin Tool is invoked. This type of exit is sometimes referred to as a BEFORE exit.

Db2 Admin Tool provides a sample invocation exit in SADBSAMP(ADBBEXIT). This exit contains a list of commonly modified Db2 Administration Tool variables. You can use this exit as a model to create your own site-specific exit.

Example of how to use the invocation exit

For this example, assume that you want to set default JOB card parameters.

The following figure is a copy of the sample invocation exit SADBSAMP(ADBBEXIT) where line 173 is updated to set the fifth line in the **job card** field on the **Batch Job Utility Parameters** panel:

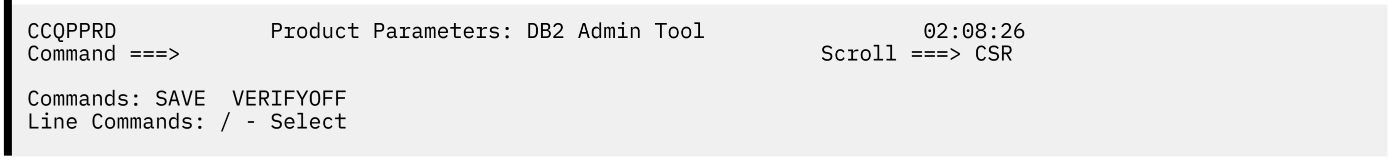


```
FLMEDDE   TS5776.EXEC(ADBBEXIT) - 01.06             Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 /*REXX*****************************************************************/
000002 /* Database 2 Administration Tool .                                    */
000003 /* 5655-DT2 Copyright IBM Corporation 2019.                            */
000004 /* All rights reserved.  Licensed materials - property of IBM.         */
000005 /* US Government Users Restricted Rights - Use, duplication or         */
000006 /* disclosure restricted by GSA ADP schedule contract with IBM Corp.  */
000007 /*                                                                    */
000008 /* Licensed Materials - Property of Rocket Software, Inc.             */
000009 /* Copyright Rocket Software, Inc. 2019.       All Rights Reserved.   */
000010 /*--------------------------------------------------------------------*/
000011 /* Exec name     : ADBBEXIT                                           */
000012 /* Purpose       : This sample Invocation Exit can be used to set     */
000013 /*                 specific DB2 Admin Tool variables site-wide to all */
000014 /*                 users. This exit or a copy of it can be specified  */
000015 /*                 as a fully-qualified data set with member name for */
000016 /*                 the following input field on IBM Tools Customizer's*/
000017 /*                 Product Parameters or DB2 Parameters panel:        */
000018 /*                     Invocation Exit . . . '<hlq>.exec(ADBBEXIT)'   */
000019 /*                                                                    */
000020 /* Instructions/guidelines on how to customize this exit:              */
000021 /* - Only customize the variable settings in subroutine ADMVARS.      */
000022 /* - In ADMVARS, uncomment any of the DB2 Admin Tool variables and    */
000023 /*   assign it a valid value.                                         */
000024 /*                                                                    */
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -   144 Line(s) not Displayed
000169   /*-----------------------------------------------------------------*/
000170   /* Jobcard line #5.                                                 */
000171   /* Value (example):                                                 */
000172   /*-----------------------------------------------------------------*/
000173   ADBJ5 = "//* TEST INVOCATION EXIT"
000174
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 30 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 21. Sample invocation exit

To use this sample invocation exit to set this JOB card information:

1. Start Tools Customizer. For detailed instructions, see Starting Tools Customizer (IBM Tools Customizer for z/OS 1.1)
2. On the **Customizer Workplace** panel, specify the E line command next to the **Product parameters** field and press Enter.
3. On the **Product Parameters** panel, in the **Invocation Exit** field, specify the qualified name of the data set member that contains the REXX exec:



```
CCQPPRD           Product Parameters: DB2 Admin Tool              02:08:26
Command ===>                                              Scroll ===> CSR

Commands: SAVE  VERIFYOFF
Line Commands: / - Select
```

```
Product customization library  : TS5776.TCZ.CUST.A6.$RS22$.ADB1210
                                                                   More:   - +
    DB2 Sort SCNKLINK lib                                                 Add
    DB2 Sort SCNKLPA lib .                                                Add
    Masking User Exit  . .                                                Add
    Invocation Exit . 'TS5776.EXEC(ADBBEXIT)'                           >
```

4. Exit the panel to save the changes.
5. On the **Customizer Workplace** panel, generate the job by specifying the G line command next to the relevant Db2 entry and press Enter:

```
CQPWIH              Customizer Workplace: DB2 Admin Tool          Row 1 to 3 of 3
Command ===>                                                      Scroll ===>
CSR

Commands: ASSOCIATE  DISCOVER  GENERATEALL JOBLIST

Customization lib TS5776.TCZ.CUST.A6.$RS22$.ADB1210              Vol TSP122

Product and LPAR Parameter Status
  Line commands: E - Edit  B -Browse
    Product parameters . : Ready to Customize
    LPAR parameters  . . : Ready to
Customize

Associated DB2 Entries and Parameter Status
  Line commands: G -Generate jobs  E - Edit  B - Browse  C - Copy  R - Remove
  Cmd SSID GrpAttch Lvl Mode User ID Date      Status             Message
      *    *        *   *    *       *         *                  *
      L7BB  --      111 NFM  TS5776 2018/09/22 Ready to Customize
      DAA4  --      101 NFM  TS5776 2018/09/22 Ready to Customize
    G DC1Q  --      121  NFM  TS5776  2019/02/18 Ready to Customize Edited
------------------------------End of DB2 entries -----------------------------
```

6. On the **Finish Product Customization** panel, open the job for editing by entering the E line command next to the member that contains the generated job:

```
CCQPCSI                 Finish Product Customization         Row 1 to 6 of 6
Command ===>                                                  Scroll ===>
CSR
Line Commands: E - Edit  B -
Browse

    Product customization library:   TS5776.TCZ.CUST.A6.$RS22$.ADB1210
 Cmd Member   New SSID GrpAttch Template Date       Description
     *        *   *    *        *        *
-   -------- --- ---- ----     -------- ---------- -------------------------->
E   A0CUSTAK NO  DC1Q  --      ADBCUST  2019/02/18 General customization
    A2SETUAK NO  DC1Q  --      ADBSETUP 2019/02/18 Admin Tool Setup Task
    A4CMBSAK NO  DC1Q  --      ADBCMBSS 2019/02/18 Create CM Batch items
    A5BINDAK NO  DC1Q  --      ADBBIND  2019/02/18 Bind packages
    A6PLANAK NO  DC1Q  --      ADBPLANS 2019/02/18 Bind plans
    B2CMBIAK NO  DC1Q  --      ADBCMBIV 2019/02/18 Verify CM Batch JCL procedu
--------------------------- End of customized jobs ----------------------------
```

7. Submit the job:

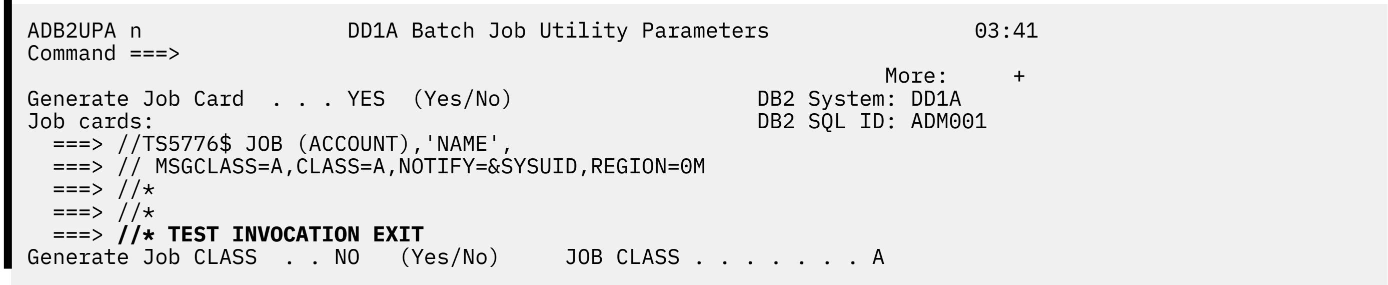
```
File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test
Help

ISREDDE2   TS5776.TCZ.CUST.A6.$RS22$.ADB1210(A0CUSTAK) - 0 Columns 00001 00072
Command ===> SUB                                              Scroll ===> CSR
000333 * INVOCATION EXIT
000334 :ADBBEXIT.'TS5776.EXEC(BEXIT)'
000335
000336
000337 * COPY FIXED-BLOCKED (FB) CLIST/EXEC LIBRARIES TO VARIABLE-BLOCKED (VB)
000338 * ---------------------------------------------------------------------
000339 * FIXED TO VARIABLE BLOCKED VOLSER
000340 :VLSRNM.
000341
000342 * FIXED TO VARIABLE BLOCKED UNIT
000343 :F2VDASD.
000344
```

8. Confirm that the JOB card variable was set:

a. Invoke Db2 Admin Tool.

b. From the main menu, specify P.BP to navigate to the **Batch Job Utility Parameters** panel.

c. Confirm that the fifth line in the **job card** field was set by the invocation exit:



```
ADB2UPA n                DD1A Batch Job Utility Parameters                03:41
Command ===>
                                                                   More:     +
Generate Job Card  . . . YES  (Yes/No)                        DB2 System: DD1A
Job cards:                                                    DB2 SQL ID: ADM001
  ===> //TS5776$ JOB (ACCOUNT),'NAME',
  ===> // MSGCLASS=A,CLASS=A,NOTIFY=&SYSUID,REGION=0M
  ===> //*
  ===> //*
  ===> //* TEST INVOCATION EXIT
Generate Job CLASS  . . NO   (Yes/No)     JOB CLASS . . . . . . . A
```

The Db2 Tools Launchpad

The Db2 Tools Launchpad provides a convenient way to run all installed IBM Db2 tools by allowing you to launch them from a centralized panel. Only tools that have an ISPF interface can be launched from the Launchpad.

Related information

“Troubleshooting: The Launchpad panel is missing product settings” on page 1062

If the Discover EXEC is not used during an Db2 Admin Tool upgrade, the Launchpad panel (if used) might be missing Launchpad product settings that were established before the upgrade.

Creating the Launchpad table

To use the Db2 Tools Launchpad, you must first create an ISPF Launchpad table. This table contains an entry for each tool that you want to be able to run from Launchpad.

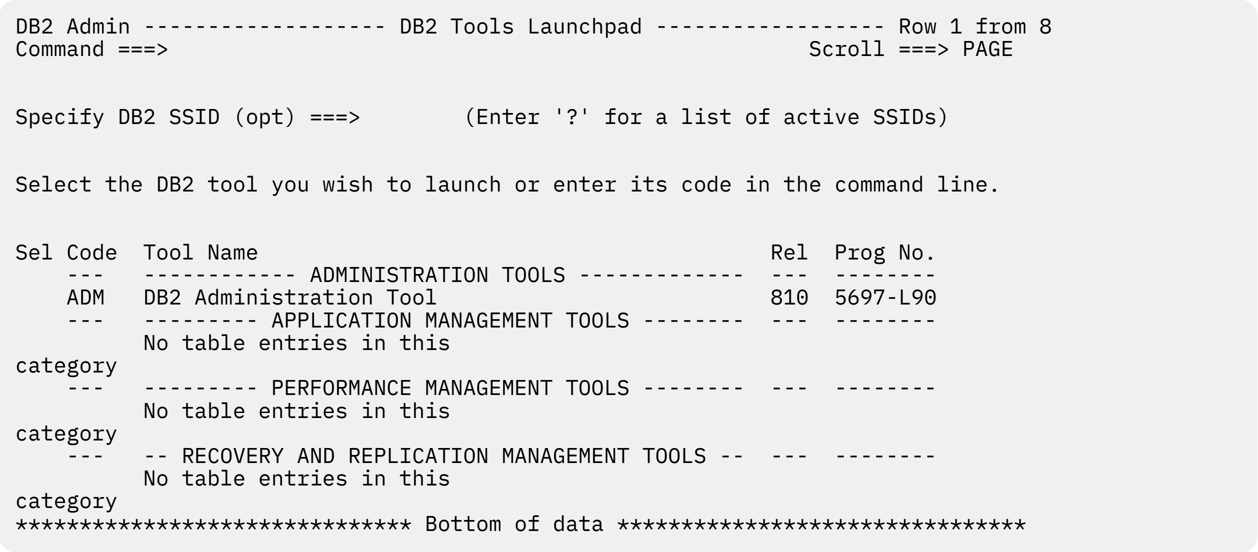
Procedure

To create the Launchpad table:

1. From a standalone TSO session or the TSO Command panel in ISPF, run the ADBL CLIST with the DMT parameter.

   The Launchpad table, ADBDMT, is created in the table library data set [the data set that is specified in the **Customized table lib** field on the **Product Parameters (CCQPPRD)** panel].

   The following figure shows the ADBDMT table immediately after it is created:



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


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


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


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

*Figure 22. **Launchpad Table (ADBDMT)** panel*

Notice that this panel groups the Db2 tools into the following four categories:

- Administration
- Application Management
- Performance Management
- Recovery and Replication Management

These categories make locating tools on the Launchpad easier.

For each tool, the panel displays the user-defined code for the product, the product name, the release or version number, and the IBM program number.

2. Optional: To make a particular Db2 subsystem available to all tools that are invoked from the Launchpad, specify the subsystem identifier (SSID) in the **Specify DB2 SSID (opt)** field. The last SSID that you specified persists across ISPF sessions.

   The SSID is stored in variable DMTSSID.

What to do next

- If you have not already done so, update the APF authorization table.
- Modify the Launchpad table.

Related tasks

"Launching tools by using the Db2 Tools Launchpad" on page 130
You can use Launchpad to quickly start IBM Db2 tools. You must have already set up the Launchpad table and modified it as needed.

Related information

"Troubleshooting: The Launchpad panel is missing product settings" on page 1062
If the Discover EXEC is not used during an Db2 Admin Tool upgrade, the Launchpad panel (if used) might be missing Launchpad product settings that were established before the upgrade.

Modifying the Launchpad table

After the Launchpad table is created, you can customize the Launchpad to include the tools you want by adding, updating, and deleting entries in the table.

Procedure

To modify the Launchpad table:

1. If the **Launchpad Table (ADBDMT)** panel is not already displayed, run the ADBL CLIST with the DMT parameter to open this panel.
2. Complete one or more of the following tasks:
   - "Adding tools to the Launchpad table" on page 128
   - "Updating tools in the Launchpad table" on page 130
   - "Deleting tools from the Launchpad table" on page 130

Adding tools to the Launchpad table

Procedure

To add tools to the Launchpad:

1. On the **Launchpad Table (ADBDMT)** panel, specify ADD in the Sel column of any row.
2. On the **DD An Entry (ADBDMTI)** panel, specify the following values to identify the tool, and press Enter:

Tool Name
The name of the Db2 tool to be displayed on the Launchpad.

Code
A user-defined short name to identify the tool.

Recommendation: Use a unique code value for each tool. Although you can use duplicate codes, Db2 could run the wrong tool when the code is invoked.

Prog No.
The IBM product program number or equivalent.

Release
The version and release numbers of the tool. For example, release number 111 indicates version 11 and release 1.

Group
Specify one of the following categories for the tool:

1
Administration Tools

2
Application Management Tools

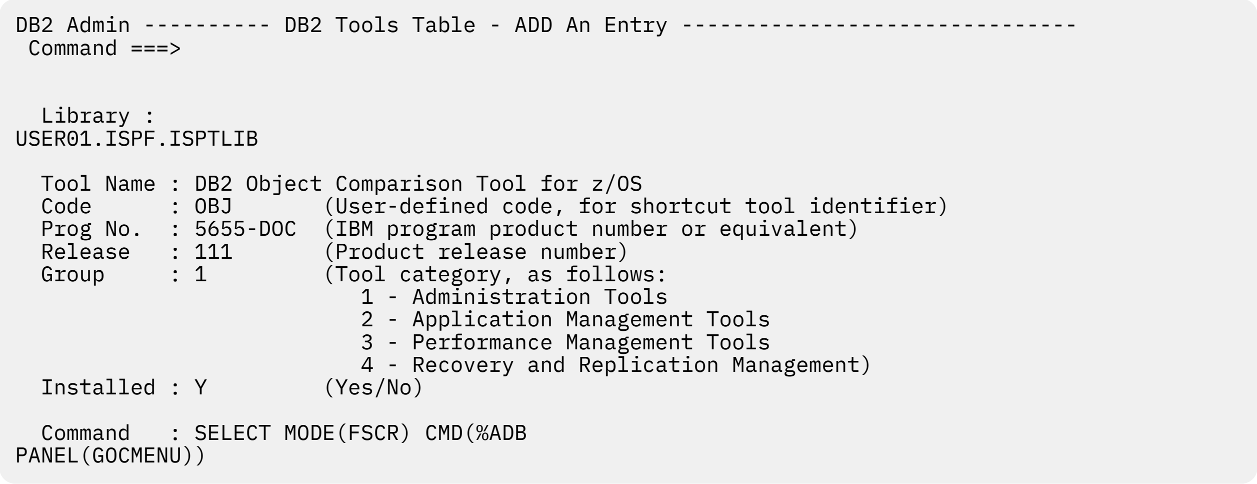
3
Performance Management Tools

4
Recovery and Replication Management Tools

Installed
Specify whether the tool is installed. Valid values are Y and N. If a tool is not installed (N), you can create a table entry for it, but the table entry is not displayed on the panel. If you install the tool later and want to include it on the Launchpad display panel, change this value to installed (Y) by following the instructions in "Updating tools in the Launchpad table" on page 130.

Command
Specify the ISPF string that is to be used to launch the tool. This field does not require continuation characters for very long command strings because it accepts a free-form format that wraps to the next line.

For additional information about these input fields, see the help panel.



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


  Library :
USER01.ISPF.ISPTLIB

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

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

*Figure 23. **DD An Entry (ADBDMTI)** panel*

The tool that you specified is added.

Updating tools in the Launchpad table

Procedure

To update tools in the Launchpad table:

1. On the **Launchpad Table (ADBDMT)** panel, specify UPD in the Sel column of any row.
2. On the **UPDATE An Entry (ADBDMTI)** panel, overwrite the information that you want to modify and press Enter.

   The entry in the Launchpad table is updated.

Deleting tools from the Launchpad table

Procedure

To delete tools from the Launchpad table:

1. On the **Launchpad Table (ADBDMT)** panel, specify DEL in the Sel column of the appropriate row.
2. On the **DELETE An Entry (ADBDMTI)** panel, press Enter.
3. Confirm whether to delete the specified tool from the table:
   - Specify Y to delete the tool.
   - Specify N or press End to cancel the delete operation.

Related tasks

"Creating the Launchpad table" on page 127
To use the Db2 Tools Launchpad, you must first create an ISPF Launchpad table. This table contains an entry for each tool that you want to be able to run from Launchpad.

"Launching tools by using the Db2 Tools Launchpad" on page 130
You can use Launchpad to quickly start IBM Db2 tools. You must have already set up the Launchpad table and modified it as needed.

Related information

"Troubleshooting: The Launchpad panel is missing product settings" on page 1062
If the Discover EXEC is not used during an Db2 Admin Tool upgrade, the Launchpad panel (if used) might be missing Launchpad product settings that were established before the upgrade.

Launching tools by using the Db2 Tools Launchpad

You can use Launchpad to quickly start IBM Db2 tools. You must have already set up the Launchpad table and modified it as needed.

Procedure

To launch tools by using the Db2 Tools Launchpad:

1. Run the ADBL CLIST with the DMT parameter.
2. On the **Launchpad Table (ADBDMT)** panel, use either of the following methods to launch a tool:
   - Specify an S or a slash (/) in the Sel column.
   - Specify the code that is associated with the tool on the command line, and press Enter.

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

   The first panel of the selected tool is displayed.

Related tasks

"Creating the Launchpad table" on page 127

To use the Db2 Tools Launchpad, you must first create an ISPF Launchpad table. This table contains an entry for each tool that you want to be able to run from Launchpad.

“Modifying the Launchpad table” on page 128
After the Launchpad table is created, you can customize the Launchpad to include the tools you want by adding, updating, and deleting entries in the table.

Related information

“Troubleshooting: The Launchpad panel is missing product settings” on page 1062
If the Discover EXEC is not used during an Db2 Admin Tool upgrade, the Launchpad panel (if used) might be missing Launchpad product settings that were established before the upgrade.

Considerations after applying a PTF

After you apply maintenance, you might need to take additional actions on your system depending on what was changed by the program temporary fix (PTF).

After applying a PTF, take the following actions as needed:

- If any of the following modules were updated by the PTF, update the load libraries for the WLM address space:
  - ADB2RE
  - ADBCSRP
  - ADB2RCP
  - ADBCRSU
  - ADBMSGT and its aliases
  - ADB0001
  - ADB1000
  - ADB3000
  - ADB9000
  - ADBGDDL

  To make this update, copy the updated module to the load library that is used by the WLM address space. This library is defined in the STEPLIB concatenation in the JCL startup procedure for the WLM address space.

Any additional actions that are required for a particular PTF are described in the ++HOLD information for that PTF.

Chapter 3. Getting started

Learn how to use the Db2 Admin Tool interface and complete a tutorial that demonstrates some of the Db2 Admin Tool capabilities.

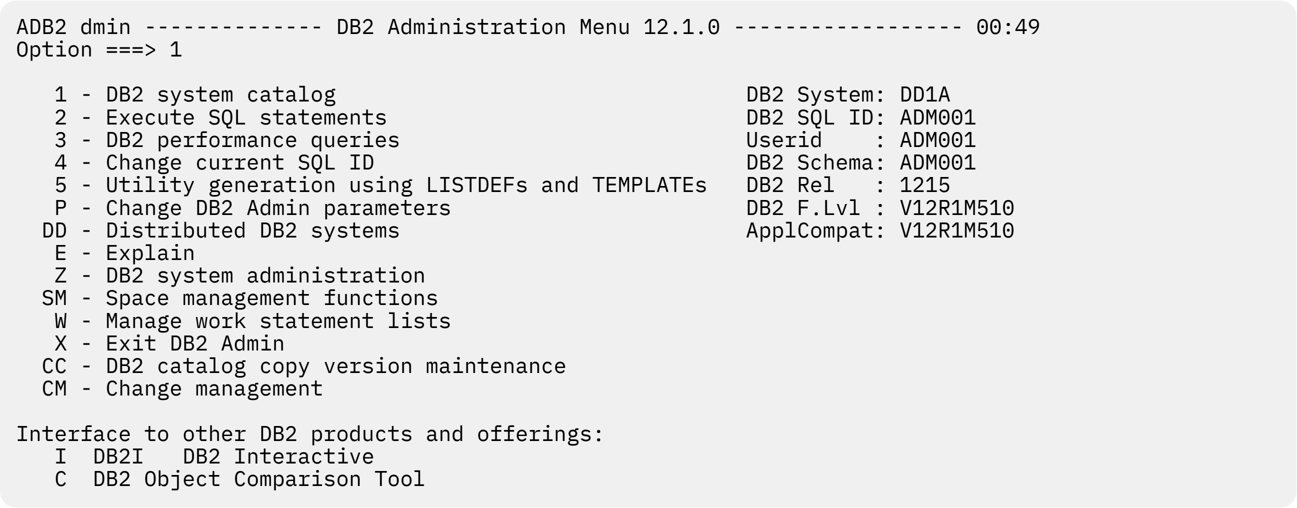
Db2 Admin Tool panels

The release level and mode or function level of your Db2 subsystems can affect the options that are available on the Db2 Admin Tool panels.

DB2 Administration Menu (ADB2) panel

The **DB2 Administration Menu (ADB2)** panel is the main menu for accessing Db2 Admin Tool functions.

Attention: You can use the TSO split screen function to access Db2 Admin Tool. However, if the Db2 subsystems that you are accessing are at different version levels, you might experience unexpected problems, such as a system abend 0C4, ABEND0C4. To avoid these problems, ensure that the different Db2 subsystems are at the same version level.



```
ADB2 dmin -------------- DB2 Administration Menu 12.1.0 ------------------ 00:49
Option ===> 1

   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   : 1215
   P - Change DB2 Admin parameters                       DB2 F.Lvl : V12R1M510
  DD - Distributed DB2 systems                           ApplCompat: V12R1M510
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

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

*Figure 24. **DB2 Administration Menu (ADB2)** panel*

This panel lists the following information:

DB2 System
The Db2 subsystem name

DB2 SQL ID
The current Db2 SQL authorization ID

Userid
The user ID

DB2 Schema
The Db2 schema

DB2 Rel
The Db2 release number

DB2 F.Lvl
The Db2 function level. See "Db2 function level" on page 70.

ApplCompat
The value of the CURRENT APPLICATION COMPATIBILITY special register. This special register is initially set to the APPLCOMPAT value with which the ADBMAIN package was bound. However, you

can update this value within Db2 Admin Tool by using a SET statement or the APPLCOMPAT primary command. See APPLCOMPAT.

From this panel, you can specify the following options:

1
Select this option to display information from the Db2 catalog about Db2 objects and authorizations for those objects.

2
Select this option to execute SQL statements.

3
Select this option to run performance and space usage queries.

4
Select this option to change your current SQL ID. This action is equivalent to issuing the Db2 Admin Tool primary command SQLID.

5
Select this option to generate utility jobs that use lists (defined by the LISTDEF utility) and templates (defined by the TEMPLATE utility).

P
Select this option to change Db2 Admin Tool parameters.

DD
Select this option to see the system catalog panels for a remote Db2 system.

E
Select this option to use the EXPLAIN statement. You can enter an SQL statement and see the resulting rows in a plan table (PLAN_TABLE), list rows from a plan table and see how Db2 will execute SQL statements in application plans or packages that were bound with EXPLAIN(YES), or create and upgrade a plan table.

Z
Select this option to display a list of system administration functions.

SM
Select this option to perform space manager functions.

W
Select this option to display the work statement list library and to manage work statement lists.

CC
Select this option to maintain and update the catalog copy version table. This option is displayed only if you customized your system to support multiple catalog copies.

CM
Select this option to use the Change Management functions. You can manage objects such as changes, versions, masks, and ignores. You can also complete tasks such as managing report changes. This option is displayed only if Db2 Admin Tool has been customized to enable the use of Change Management.

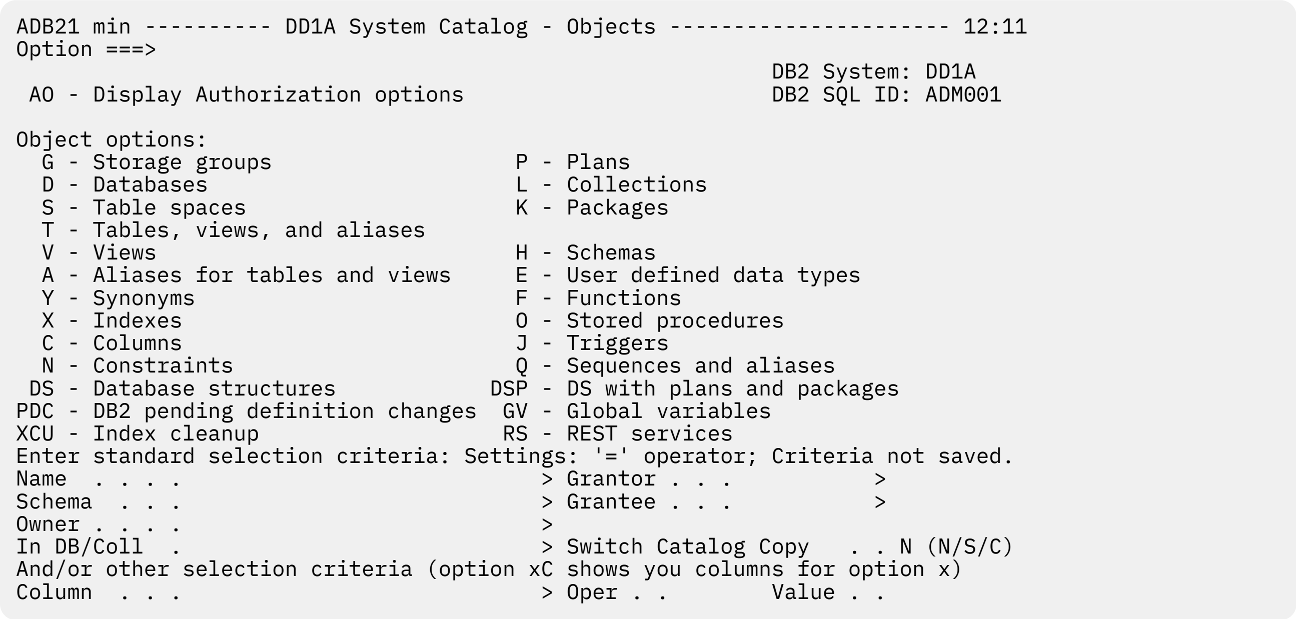
S
This option is “hidden”, meaning that it is not displayed on the panel. Select **S** to access the Db2 Admin Tool product sample application.

System catalog panels

The main system catalog panels are described in this reference information.

The System Catalog (ADB21) panel

Use the **System Catalog (ADB21)** panel to search for Db2 objects or authorizations in the Db2 catalog. To display this panel, select option 1 on the **DB2 Administration Menu (ADB2)** panel.



```
ADB21 min ---------- DD1A System Catalog - Objects ---------------------- 12:11
Option ===>
                                                           DB2 System: DD1A
 AO - Display Authorization options                        DB2 SQL ID: ADM001

Object options:
  G - Storage groups                   P - Plans
  D - Databases                        L - Collections
  S - Table spaces                     K - Packages
  T - Tables, views, and aliases
  V - Views                            H - Schemas
  A - Aliases for tables and views     E - User defined data types
  Y - Synonyms                         F - Functions
  X - Indexes                          O - Stored procedures
  C - Columns                          J - Triggers
  N - Constraints                      Q - Sequences and aliases
 DS - Database structures            DSP - DS with plans and packages
PDC - DB2 pending definition changes  GV - Global variables
XCU - Index cleanup                   RS - REST services
Enter standard selection criteria: Settings: '=' operator; Criteria not saved.
Name  . . . .                            > Grantor . . .           >
Schema  . . .                            > Grantee . . .           >
Owner . . . .                            >
In DB/Coll  .                            > Switch Catalog Copy   . . N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .                            > Oper . .        Value . .
```

Figure 25. ***System Catalog (ADB21)*** *panel*

On the **System Catalog (ADB21)** panel, enter one of the object codes on the command line (for example, D for databases). You can limit the information that is returned by entering one or more selection criteria at the bottom of the panel. For example, specifying D402 in the **Name** field limits the search to databases whose names begin with D402. When you press Enter, Db2 Admin Tool creates and executes an SQL statement that searches the Db2 catalog for the object or authorization that you have requested.

Filtering tips

One way to filter your selection is to use the **In DB/Coll** (database or collection) field. For example, if you want to display table spaces within a specific database, select option S and enter the name of a database in the **In DB/Coll** field. Or, if you want to display a specific collection in a package, select option K and specify the collection ID in the **In DB/Coll** field.

Recommendation: For optimum performance when you specify option T, specify selection criteria. Enter a value for **Owner** or **In DB/Coll**.

Exact and fuzzy searches

When you specify selection criteria, you can change from a LIKE search (also known as a "fuzzy" search) to an exact search, which uses an equal sign (=). Use the LIKE ON and LIKE OFF primary commands to toggle between a "fuzzy" search (LIKE ON) and an exact search (LIKE OFF).

They type of search used is shown in the `Enter standard selection criteria` line on the panel. For example, the following line indicates that a LIKE search ("fuzzy" search) is used:

```
Enter standard selection criteria: Settings: LIKE operator; Criteria not saved.
```

Saving search criteria

You can save (or not save) your search criteria between Db2 Admin Tool sessions by using the SAVE ON and SAVE OFF primary commands.

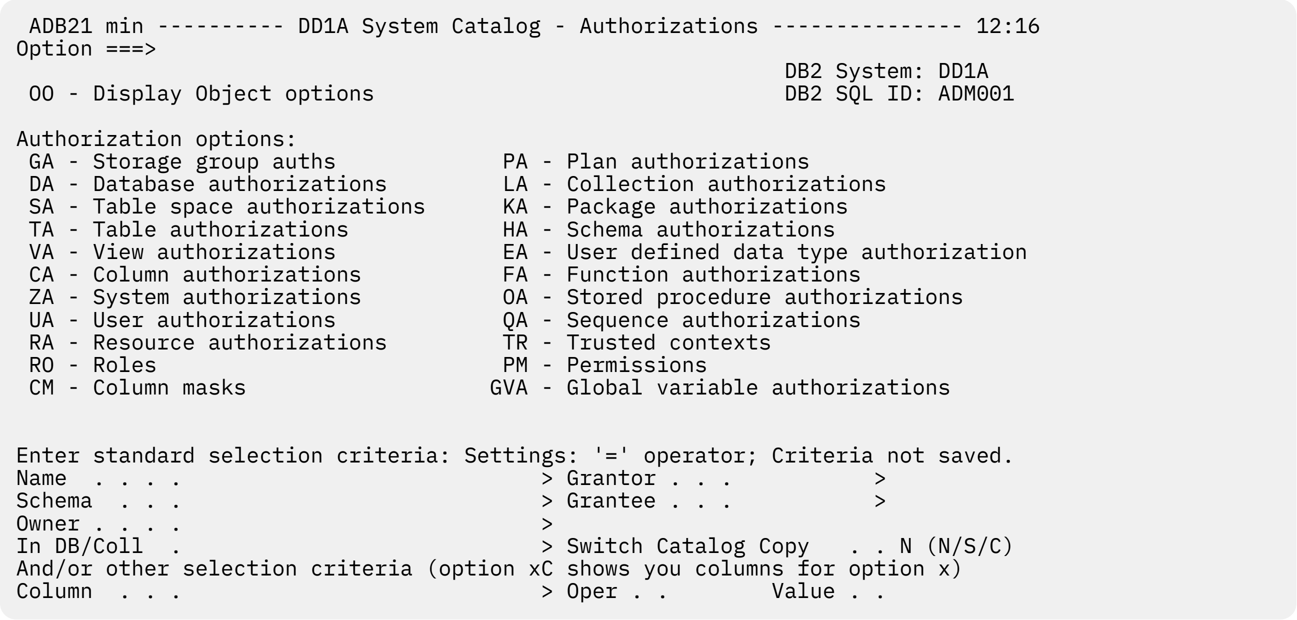
Whether search criteria is saved is shown in the `Enter standard selection criteria` line on the panel. For example, the following line indicates that the criteria is saved (SAVE ON):

```
Enter standard selection criteria: Settings: LIKE operator; Criteria saved.
```

With SAVE ON, the search criteria is restored when you re-enter a Db2 Admin Tool session.

Authorization options

To view the authorization options, choose the AO option on the **System Catalog (ADB21)** panel. The authorization options are then displayed on the **System Catalog (ADB21)** panel - Authorization options.



```
 ADB21 min ---------- DD1A System Catalog - Authorizations --------------- 12:16
Option ===>
                                                            DB2 System: DD1A
 OO - Display Object options                                DB2 SQL ID: ADM001

Authorization options:
 GA - Storage group auths             PA - Plan authorizations
 DA - Database authorizations         LA - Collection authorizations
 SA - Table space authorizations      KA - Package authorizations
 TA - Table authorizations            HA - Schema authorizations
 VA - View authorizations             EA - User defined data type authorization
 CA - Column authorizations           FA - Function authorizations
 ZA - System authorizations           OA - Stored procedure authorizations
 UA - User authorizations             QA - Sequence authorizations
 RA - Resource authorizations         TR - Trusted contexts
 RO - Roles                           PM - Permissions
 CM - Column masks                   GVA - Global variable authorizations


Enter standard selection criteria: Settings: '=' operator; Criteria not saved.
Name  . . . .                            > Grantor . . .           >
Schema  . . .                            > Grantee . . .           >
Owner . . . .                            >
In DB/Coll  .                            > Switch Catalog Copy   . . N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .                            > Oper . .        Value . .
```

*Figure 26. **System Catalog (ADB21)** panel - Authorization options*

For optimum performance, specify selection criteria for all authorization options (*x*A) and enter a value for **Grantor** or **Grantee**.

Db2 Admin Tool reports authorizations based solely on the Db2 catalog. However, the actual authorization is affected by other sources that are external to the Db2 catalog, such as the following sources:

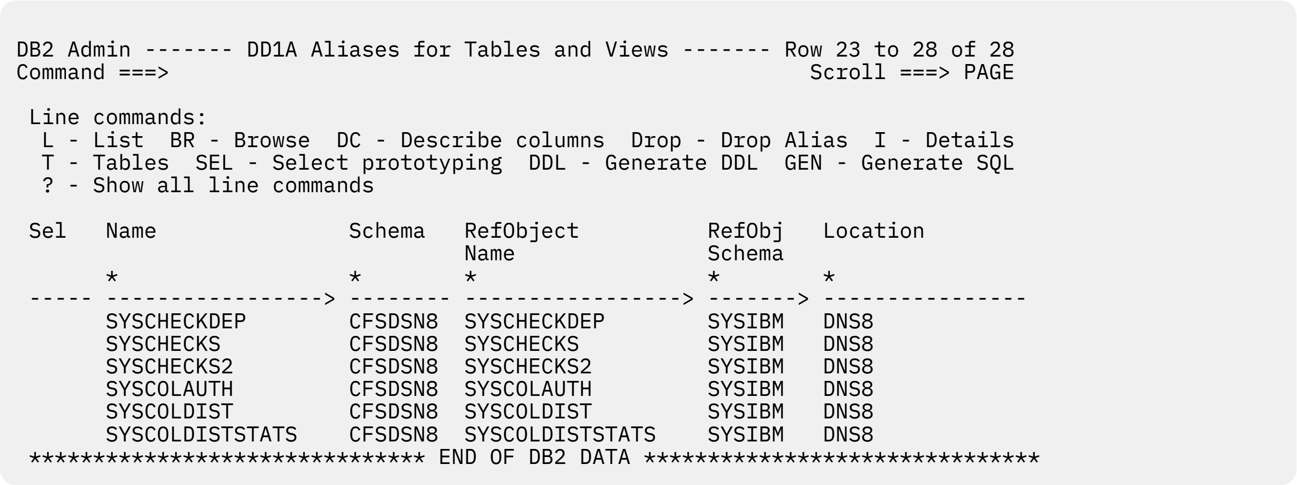
- Installation authorities that are specified in DSNZPARM.
- Any external security system, such as the RACF program.
- Any security product from another software provider.
- Security user exits, including the ones that are supplied by IBM.

Option A. Aliases

Use the **Aliases** panel to display information about the aliases in the Db2 catalog.

Select option A on the **System Catalog** panel to display the **Aliases** panel, as shown in the following figure.

On this panel, you can reverse engineer Db2 objects by using the GEN line command.



```
DB2 Admin ------- DD1A Aliases for Tables and Views ------- Row 23 to 28 of 28
Command ===>                                                  Scroll ===> PAGE

 Line commands:
  L - List  BR - Browse  DC - Describe columns  Drop - Drop Alias  I - Details
  T - Tables  SEL - Select prototyping  DDL - Generate DDL  GEN - Generate SQL
  ? - Show all line commands

 Sel   Name               Schema   RefObject          RefObj   Location
                                   Name               Schema
       *                  *        *                  *        *
 ----- -----------------> -------- -----------------> -------> ----------------
       SYSCHECKDEP        CFSDSN8  SYSCHECKDEP        SYSIBM   DNS8
       SYSCHECKS          CFSDSN8  SYSCHECKS          SYSIBM   DNS8
       SYSCHECKS2         CFSDSN8  SYSCHECKS2         SYSIBM   DNS8
       SYSCOLAUTH         CFSDSN8  SYSCOLAUTH         SYSIBM   DNS8
       SYSCOLDIST         CFSDSN8  SYSCOLDIST         SYSIBM   DNS8
       SYSCOLDISTSTATS    CFSDSN8  SYSCOLDISTSTATS    SYSIBM   DNS8
 ******************************* END OF DB2 DATA *******************************
```

Figure 27. The Aliases panel (ADB21A) – displaying aliases

The following columns are on this panel:

Sel
Input field where you enter one of the line commands listed on the panel.

Name
Name of the alias.

Owner
Authorization ID of the owner of the alias.

RefObject Name
Name of the table or view to which the alias refers.

RefObj Schema
The schema of the table or view to which the alias refers.

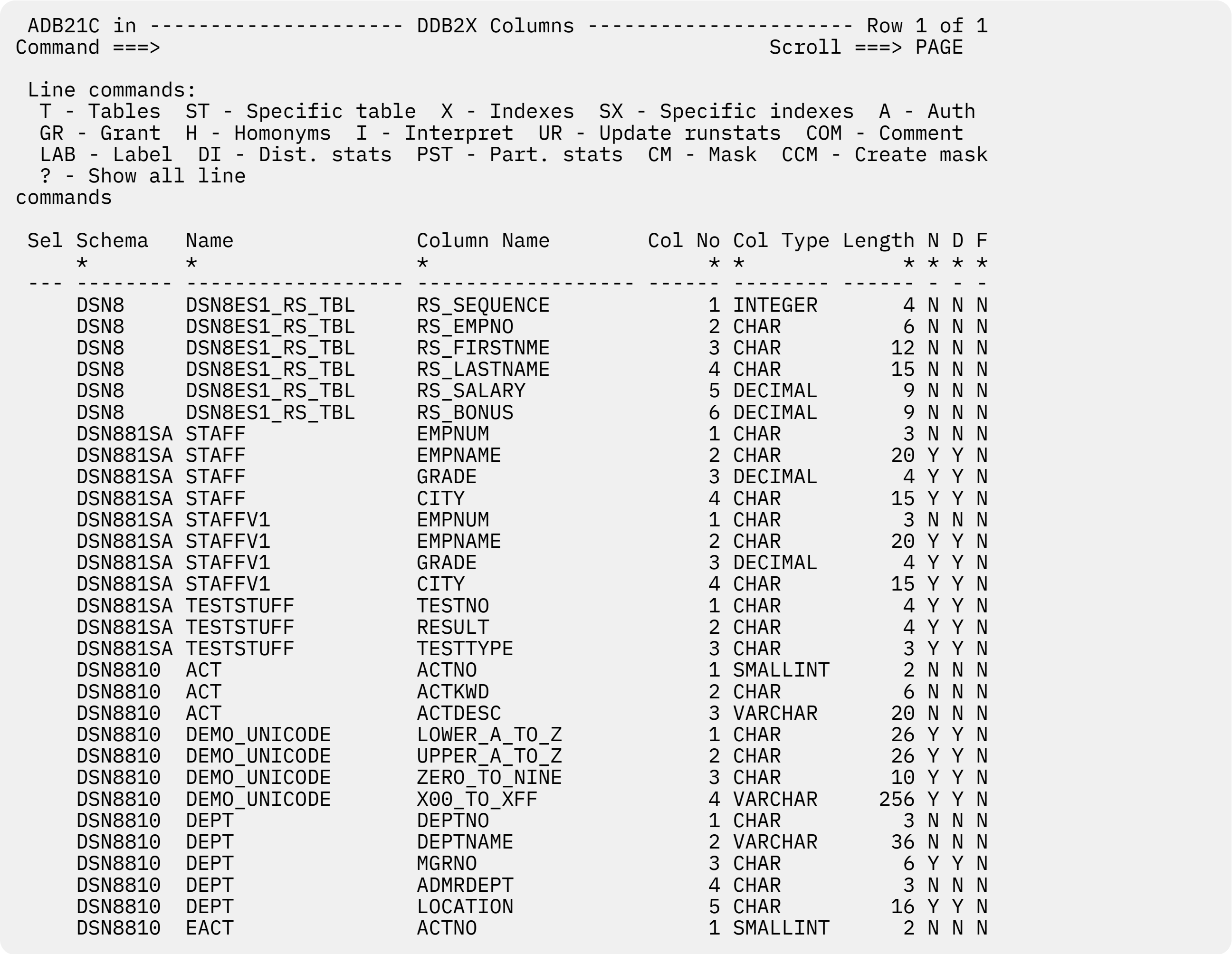
Location
Location name of the object of the alias. The field is blank for an alias that was not defined with a three-part object name.

Option C. Columns

The **Columns** panel displays the columns in the Db2 catalog.

Select option C on the **System Catalog** panel (see "The System Catalog (ADB21) panel" on page 135) to display the **Columns** panel.

The following figure shows the **Columns** panel.



```
 ADB21C in -------------------- DDB2X Columns --------------------- Row 1 of 1
Command ===>                                                 Scroll ===> PAGE

 Line commands:
  T - Tables  ST - Specific table  X - Indexes  SX - Specific indexes  A - Auth
  GR - Grant  H - Homonyms  I - Interpret  UR - Update runstats  COM - Comment
  LAB - Label  DI - Dist. stats  PST - Part. stats  CM - Mask  CCM - Create mask
  ? - Show all line
commands

 Sel Schema   Name               Column Name        Col No Col Type Length N D F
     *        *                  *                       * *             * * * *
 --- -------- ------------------ ------------------ ------ -------- ------ - - -
     DSN8     DSN8ES1_RS_TBL     RS_SEQUENCE             1 INTEGER       4 N N N
     DSN8     DSN8ES1_RS_TBL     RS_EMPNO                2 CHAR          6 N N N
     DSN8     DSN8ES1_RS_TBL     RS_FIRSTNME             3 CHAR         12 N N N
     DSN8     DSN8ES1_RS_TBL     RS_LASTNAME             4 CHAR         15 N N N
     DSN8     DSN8ES1_RS_TBL     RS_SALARY               5 DECIMAL       9 N N N
     DSN8     DSN8ES1_RS_TBL     RS_BONUS                6 DECIMAL       9 N N N
     DSN881SA STAFF              EMPNUM                  1 CHAR          3 N N N
     DSN881SA STAFF              EMPNAME                 2 CHAR         20 Y Y N
     DSN881SA STAFF              GRADE                   3 DECIMAL       4 Y Y N
     DSN881SA STAFF              CITY                    4 CHAR         15 Y Y N
     DSN881SA STAFFV1            EMPNUM                  1 CHAR          3 N N N
     DSN881SA STAFFV1            EMPNAME                 2 CHAR         20 Y Y N
     DSN881SA STAFFV1            GRADE                   3 DECIMAL       4 Y Y N
     DSN881SA STAFFV1            CITY                    4 CHAR         15 Y Y N
     DSN881SA TESTSTUFF          TESTNO                  1 CHAR          4 Y Y N
     DSN881SA TESTSTUFF          RESULT                  2 CHAR          4 Y Y N
     DSN881SA TESTSTUFF          TESTTYPE                3 CHAR          3 Y Y N
     DSN8810  ACT                ACTNO                   1 SMALLINT      2 N N N
     DSN8810  ACT                ACTKWD                  2 CHAR          6 N N N
     DSN8810  ACT                ACTDESC                 3 VARCHAR      20 N N N
     DSN8810  DEMO_UNICODE       LOWER_A_TO_Z            1 CHAR         26 Y Y N
     DSN8810  DEMO_UNICODE       UPPER_A_TO_Z            2 CHAR         26 Y Y N
     DSN8810  DEMO_UNICODE       ZERO_TO_NINE            3 CHAR         10 Y Y N
     DSN8810  DEMO_UNICODE       X00_TO_XFF              4 VARCHAR     256 Y Y N
     DSN8810  DEPT               DEPTNO                  1 CHAR          3 N N N
     DSN8810  DEPT               DEPTNAME                2 VARCHAR      36 N N N
     DSN8810  DEPT               MGRNO                   3 CHAR          6 Y Y N
     DSN8810  DEPT               ADMRDEPT                4 CHAR          3 N N N
     DSN8810  DEPT               LOCATION                5 CHAR         16 Y Y N
     DSN8810  EACT               ACTNO                   1 SMALLINT      2 N N N
```

Figure 28. Columns panel (ADB21C)

The fields on this panel are:

Sel
Input field where you enter one of the line commands listed on the panel.

Schema
Schema of the table or view that contains the column.

Name
Name of the table or view that contains the column.

Column Name
Name of the column.

Col No
Numerical position of the column in the table or view.

Col Type
Type of column, which is one of the following data types:

INTEGER
Large integer.

SMALLINT
Small integer.

FLOAT
Floating-point.

CHAR
Fixed-length character string.

VARCHAR
Varying-length character string.

LONGVAR
Varying-length character string.

DECIMAL
Decimal.

GRAPHIC
Fixed-length graphic string.

VARG
Varying-length graphic string.

LONGVARG
Varying-length graphic string.

DATE
Date.

TIME
Time.

TIMESTMP
Time stamp.

BLOB
Binary large object.

CLOB
Character large object.

DBCLOB
Double-byte character large object.

ROWID
Row ID data type.

DISTINCT
Distinct type.

Length
Length attribute of the column or, in the case of a decimal column, its precision. The number does not include internal prefixes to record actual length and null state (where these are applicable).

N
This field indicates whether the column can contain null values. This field contains one of the following values:

Y
Yes.

N
No.

D
Default value for the column. This field contains one of the following values:

N
None.

Y
Yes.

B
Yes.

1–6
User-defined defaults.

S
SQLID.

U
USER.

A
Generated always.

D
Generated by default.

I
As identity and generated always.

J
As identity and generated as default.

F
This field indicates whether the column has a field procedure. This field contains one of the following values:

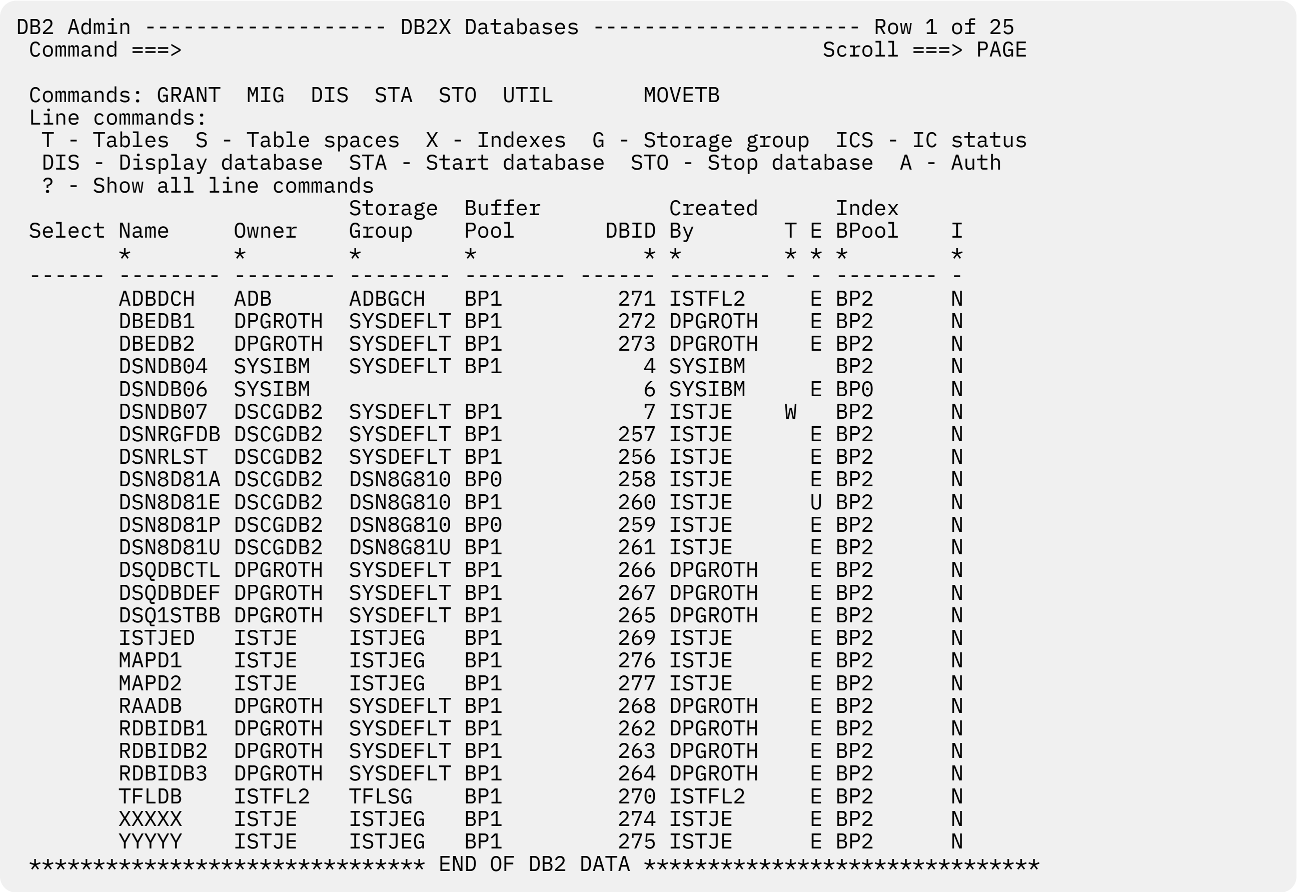
Y
Yes.

N
No.

Option D. Databases

The **Databases (ADB21D)** panel displays the databases in the Db2 catalog.

To display the **Databases (ADB21D)** panel, select option D on the **System Catalog (ADB21)** panel .



```
DB2 Admin ------------------- DB2X Databases --------------------- Row 1 of 25
 Command ===>                                                  Scroll ===> PAGE

 Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
 Line commands:
  T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
  DIS - Display database  STA - Start database  STO - Stop database  A - Auth
  ? - Show all line commands
                          Storage  Buffer          Created      Index
 Select Name     Owner    Group    Pool       DBID By       T E BPool    I
        *        *        *        *             * *        * * *        *
 ------ -------- -------- -------- -------- ------ -------- - - -------- -
        ADBDCH   ADB      ADBGCH   BP1         271 ISTFL2     E BP2      N
        DBEDB1   DPGROTH  SYSDEFLT BP1         272 DPGROTH    E BP2      N
        DBEDB2   DPGROTH  SYSDEFLT BP1         273 DPGROTH    E BP2      N
        DSNDB04  SYSIBM   SYSDEFLT BP1           4 SYSIBM       BP2      N
        DSNDB06  SYSIBM                          6 SYSIBM     E BP0      N
        DSNDB07  DSCGDB2  SYSDEFLT BP1           7 ISTJE    W   BP2      N
        DSNRGFDB DSCGDB2  SYSDEFLT BP1         257 ISTJE      E BP2      N
        DSNRLST  DSCGDB2  SYSDEFLT BP1         256 ISTJE      E BP2      N
        DSN8D81A DSCGDB2  DSN8G810 BP0         258 ISTJE      E BP2      N
        DSN8D81E DSCGDB2  DSN8G810 BP1         260 ISTJE      U BP2      N
        DSN8D81P DSCGDB2  DSN8G810 BP0         259 ISTJE      E BP2      N
        DSN8D81U DSCGDB2  DSN8G81U BP1         261 ISTJE      E BP2      N
        DSQDBCTL DPGROTH  SYSDEFLT BP1         266 DPGROTH    E BP2      N
        DSQDBDEF DPGROTH  SYSDEFLT BP1         267 DPGROTH    E BP2      N
        DSQ1STBB DPGROTH  SYSDEFLT BP1         265 DPGROTH    E BP2      N
        ISTJED   ISTJE    ISTJEG   BP1         269 ISTJE      E BP2      N
        MAPD1    ISTJE    ISTJEG   BP1         276 ISTJE      E BP2      N
        MAPD2    ISTJE    ISTJEG   BP1         277 ISTJE      E BP2      N
        RAADB    DPGROTH  SYSDEFLT BP1         268 DPGROTH    E BP2      N
        RDBIDB1  DPGROTH  SYSDEFLT BP1         262 DPGROTH    E BP2      N
        RDBIDB2  DPGROTH  SYSDEFLT BP1         263 DPGROTH    E BP2      N
        RDBIDB3  DPGROTH  SYSDEFLT BP1         264 DPGROTH    E BP2      N
        TFLDB    ISTFL2   TFLSG    BP1         270 ISTFL2     E BP2      N
        XXXXX    ISTJE    ISTJEG   BP1         274 ISTJE      E BP2      N
        YYYYY    ISTJE    ISTJEG   BP1         275 ISTJE      E BP2      N
 ******************************* END OF DB2 DATA ********************************
```

Figure 29. ***Databases (ADB21D)*** *panel*

The following primary commands are valid on this panel:

GRANT
Issues a GRANT command on multiple databases.

MIG
Issues a MIG command on multiple databases.

DIS
Issues a Db2 DISPLAY command on multiple databases.

STA
Issues a Db2 START command on multiple databases.

STO
Issues a Db2 STOP command on multiple databases.

UTIL
Selects the table spaces for multiple databases for which to generate utility JCL.

MOVETB
Moves tables from multi-table table spaces to partition-by-growth (PBG) universal table spaces (UTS).

If either of the following conditions are true, you are prompted to send the statements to a batch job or work statement list (WSL):

- The size of the statements generated by the GRANT, DIS, STA, or STO primary command exceeds 32K (an ISPF limit).
- The number of statements generated by the DIS, STA, or STO primary command exceeds 10

Recommendation: Primary commands operate on each row that is displayed in the table. If you want to omit some of the rows before you issue the primary command, use the minus (-) line command to remove rows from the display. The primary commands operate only on rows that are listed.

The panel displays the following columns:

Select
Input field where you enter one of the line commands listed on the panel.

Name
Name of the database.

Owner
Authorization ID of the owner of the database.

Storage Group
Name of the default storage group for the database. For system databases, this field is blank.

Buffer Pool
Name of the default buffer pool for the database. For system databases, this field is blank.

DBID
Internal ID for the database.

Created By
Primary authorization ID of the user who created the database.

T
Type of database. This column can contain one of the following values:

W
Work file

T
Temporary database

blank
Not a work file database or a temporary database

E
Type of encoding. This column can contain one of the following values:

E
EBCDIC

A
ASCII

U
Unicode

blank
Work file or temporary database

Index BPool
Name of the default buffer pool for indexes.

I
Whether the database is an implicitly-created database.

Y
Yes

N
No

Option DS. Database Structures

When you select option DS, the **Database Structures** panel displays a structured list of objects in the database that you have selected but does not display plans and packages.

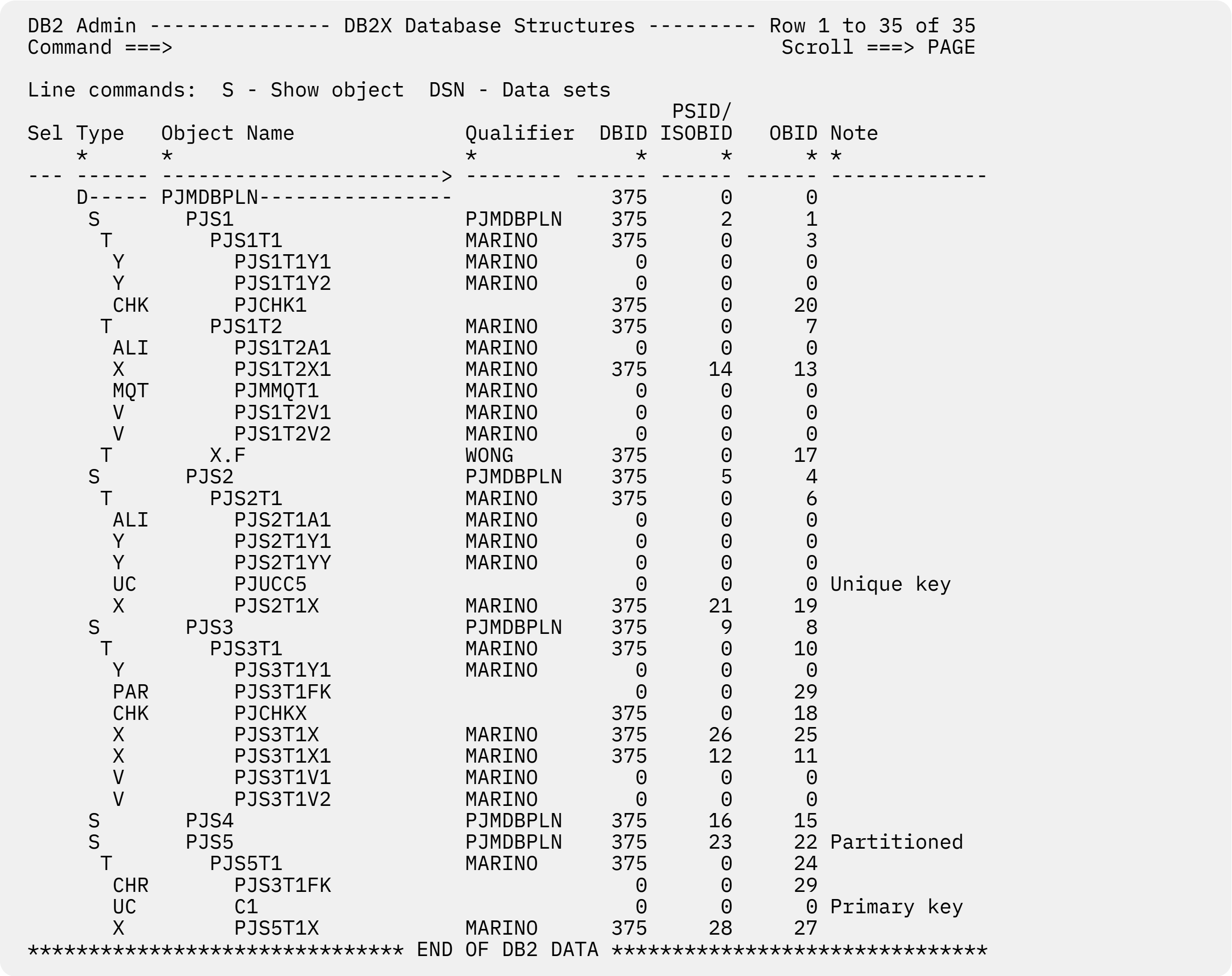
Select option DS on the **System Catalog** panel to display the **Database Structures** panel, as shown in the following figure. You must enter a value in the Name field prior to selecting the DS option. Otherwise, you will receive the following message: `Invalid for this option`.

The following object types are displayed on the **Database Structures** panel:

- Databases
- Table spaces
- Tables
- Materialized query tables
- Indexes
- Aliases
- Views on a table
- Synonyms on a table
- Triggers
- Check conditions
- Unique constraints
- Referential constraints (parents)
- Referential constraints (children)

Views on a view and authorizations are not included in this display.

The following figure shows the **Database Structures** panel without plans and packages displayed.



```
DB2 Admin --------------- DB2X Database Structures --------- Row 1 to 35 of 35
Command ===>                                                  Scroll ===> PAGE

Line commands:  S - Show object  DSN - Data sets
                                                     PSID/
Sel Type   Object Name              Qualifier  DBID ISOBID   OBID Note
    *      *                        *             *      *      * *
--- ------ ----------------------> -------- ------ ------ ------ -------------
    D----- PJMDBPLN----------------             375      0      0
     S       PJS1                   PJMDBPLN    375      2      1
      T        PJS1T1               MARINO      375      0      3
       Y         PJS1T1Y1           MARINO        0      0      0
       Y         PJS1T1Y2           MARINO        0      0      0
       CHK       PJCHK1                         375      0     20
      T        PJS1T2               MARINO      375      0      7
       ALI       PJS1T2A1           MARINO        0      0      0
       X         PJS1T2X1           MARINO      375     14     13
       MQT       PJMMQT1            MARINO        0      0      0
       V         PJS1T2V1           MARINO        0      0      0
       V         PJS1T2V2           MARINO        0      0      0
      T        X.F                  WONG        375      0     17
     S       PJS2                   PJMDBPLN    375      5      4
      T        PJS2T1               MARINO      375      0      6
       ALI       PJS2T1A1           MARINO        0      0      0
       Y         PJS2T1Y1           MARINO        0      0      0
       Y         PJS2T1YY           MARINO        0      0      0
       UC        PJUCC5                           0      0      0 Unique key
       X         PJS2T1X            MARINO      375     21     19
     S       PJS3                   PJMDBPLN    375      9      8
      T        PJS3T1               MARINO      375      0     10
       Y         PJS3T1Y1           MARINO        0      0      0
       PAR       PJS3T1FK                         0      0     29
       CHK       PJCHKX                         375      0     18
       X         PJS3T1X            MARINO      375     26     25
       X         PJS3T1X1           MARINO      375     12     11
       V         PJS3T1V1           MARINO        0      0      0
       V         PJS3T1V2           MARINO        0      0      0
     S       PJS4                   PJMDBPLN    375     16     15
     S       PJS5                   PJMDBPLN    375     23     22 Partitioned
      T        PJS5T1               MARINO      375      0     24
       CHR       PJS3T1FK                         0      0     29
       UC        C1                               0      0      0 Primary key
       X         PJS5T1X            MARINO      375     28     27
******************************* END OF DB2 DATA *******************************
```

Figure 30. Database Structures panel (ADB21DS) without plans and packages displayed

The following columns are on this panel:

Sel
Input field where you enter line command S to show an object.

Type
Type of object, which is one of the following:

ALI
Alias.

CHK
Check constraint.

CHR
Referential constraint: parent to child.

D
Database.

J
Trigger.

K
Package (shown only for the DSP command).

MQT
Materialized query table (treated as a table when preceded by two blanks in the **Type** field and as a view when preceded by three blanks).

P
Plan (shown only for the DSP command).

PAR
Referential constraint: child to parent.

S
Table space.

T
Table.

UC
Unique constraint.

V
View.

X
Index.

Y
Synonym.

Object Name
Name of the object.

Qualifier
Db2 qualifier for the object, if relevant.

DBID
Internal identifier of the database.

PSID/ISOBID
Internal identifier of the table space page set descriptor or index page set descriptor.

OBID
Identifier for the object's internal descriptor.

Option DSP. Database Structures with Plans and Packages

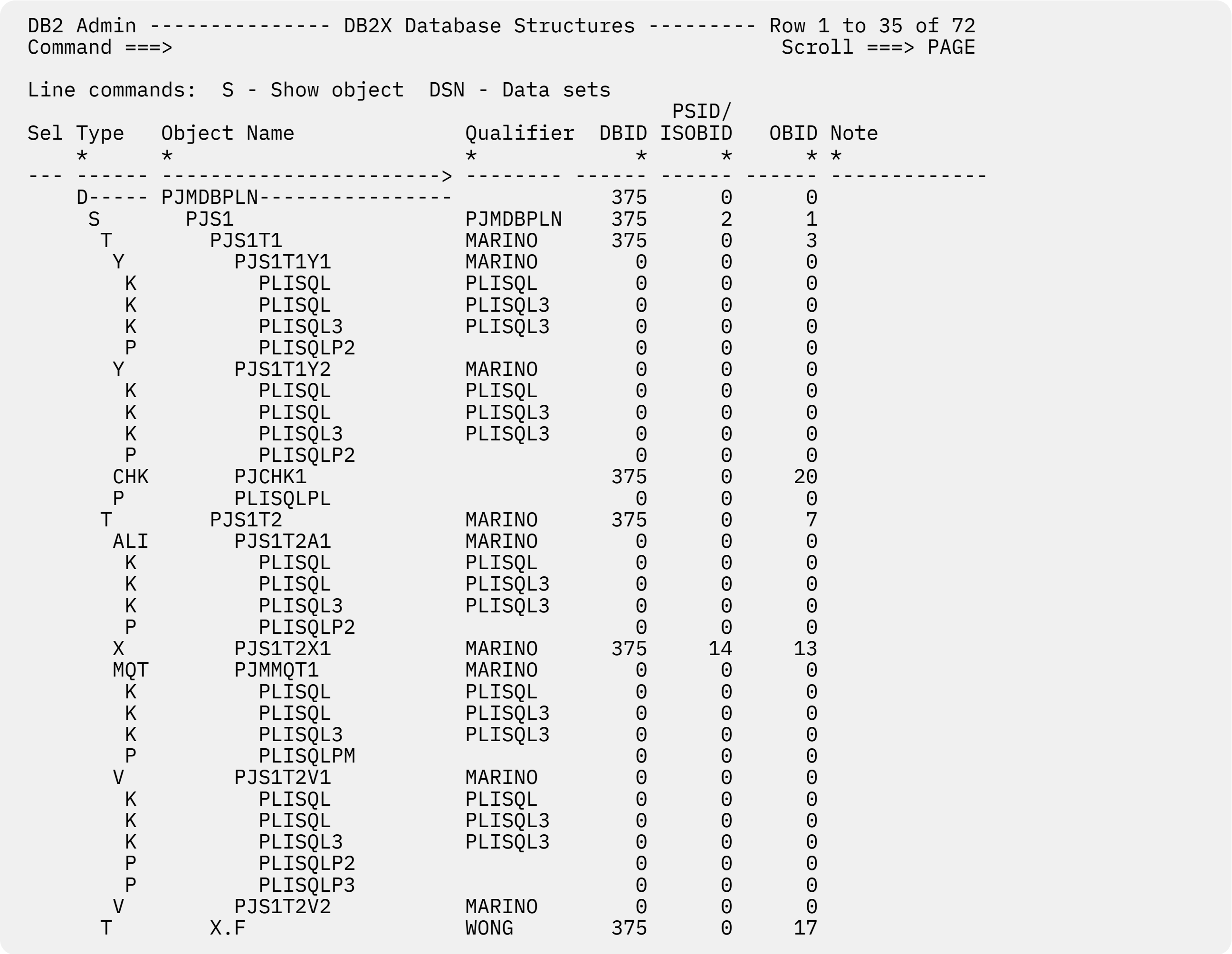
When you select option DSP, the **Database Structures** panel shows plans and packages that are dependent on the table spaces, tables, views, indexes, aliases, and synonyms.

Select option DSP on the **System Catalog** panel to display the **Database Structures** panel, as shown in the following figure, that includes showing the plans and packages that are dependent on the table spaces, tables, views, indexes, aliases, and synonyms.

In the **Database Structures** panel, plans (P) and packages (K) are indented under the object upon which they are dependent. To eliminate repetitiveness in the display, a dependency on a table is not shown if it is already reported under a view, alias, synonym, or index for the table. Likewise, a dependency for a table space is not shown if it is already reported under a table.

You must enter a value in the Name field prior to selecting the DSP option. Otherwise, you will receive the message, `Invalid for this option`.

The following figure shows the **Database Structures** panel with plans and packages displayed.



```
DB2 Admin --------------- DB2X Database Structures --------- Row 1 to 35 of 72
Command ===>                                                  Scroll ===> PAGE

Line commands:  S - Show object  DSN - Data sets
                                                     PSID/
Sel Type   Object Name              Qualifier  DBID ISOBID   OBID Note
    *      *                        *             *      *      * *
--- ------ -----------------------> -------- ------ ------ ------ ------------
    D----- PJMDBPLN----------------             375      0      0
     S       PJS1                   PJMDBPLN    375      2      1
      T        PJS1T1               MARINO      375      0      3
       Y         PJS1T1Y1           MARINO        0      0      0
        K          PLISQL           PLISQL        0      0      0
        K          PLISQL           PLISQL3       0      0      0
        K          PLISQL3          PLISQL3       0      0      0
        P          PLISQLP2                       0      0      0
       Y         PJS1T1Y2           MARINO        0      0      0
        K          PLISQL           PLISQL        0      0      0
        K          PLISQL           PLISQL3       0      0      0
        K          PLISQL3          PLISQL3       0      0      0
        P          PLISQLP2                       0      0      0
       CHK       PJCHK1                         375      0     20
       P         PLISQLPL                         0      0      0
      T        PJS1T2               MARINO      375      0      7
       ALI       PJS1T2A1           MARINO        0      0      0
        K          PLISQL           PLISQL        0      0      0
        K          PLISQL           PLISQL3       0      0      0
        K          PLISQL3          PLISQL3       0      0      0
        P          PLISQLP2                       0      0      0
       X         PJS1T2X1           MARINO      375     14     13
       MQT       PJMMQT1            MARINO        0      0      0
        K          PLISQL           PLISQL        0      0      0
        K          PLISQL           PLISQL3       0      0      0
        K          PLISQL3          PLISQL3       0      0      0
        P          PLISQLPM                       0      0      0
       V         PJS1T2V1           MARINO        0      0      0
        K          PLISQL           PLISQL        0      0      0
        K          PLISQL           PLISQL3       0      0      0
        K          PLISQL3          PLISQL3       0      0      0
        P          PLISQLP2                       0      0      0
        P          PLISQLP3                       0      0      0
       V         PJS1T2V2           MARINO        0      0      0
      T        X.F                  WONG        375      0     17
```

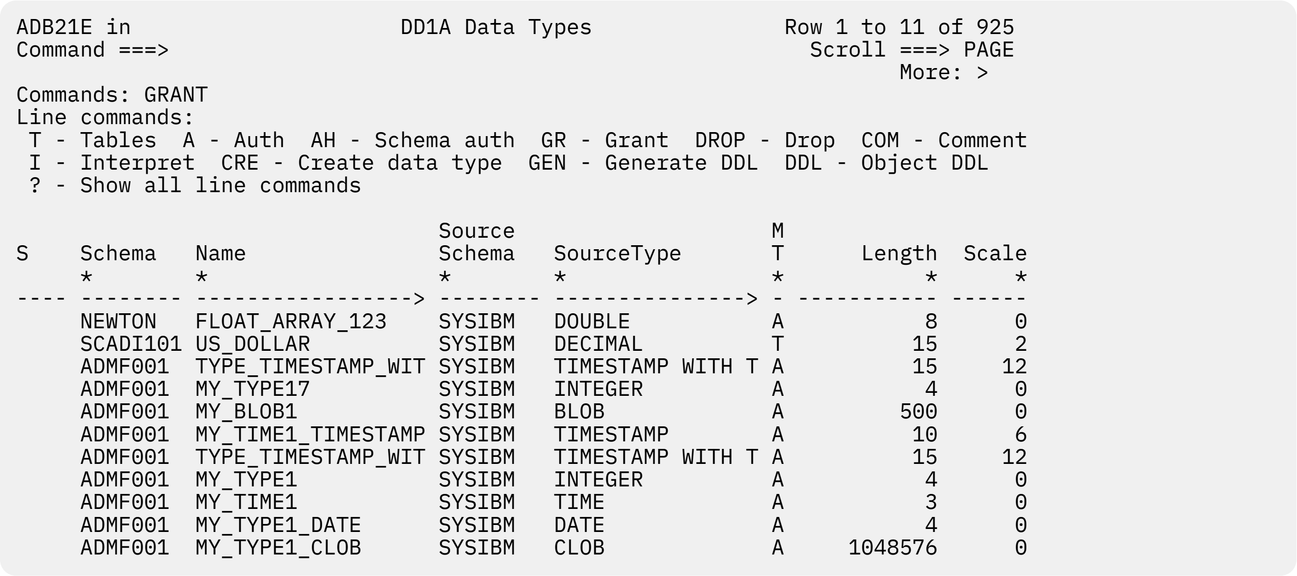
Figure 31. Database Structures panel (ADB21DS) with plans and packages displayed

Option E. User-Defined Data Types

Use the **Data Types** panel to display information about the data types in the Db2 catalog.

Select option E on the **System Catalog** panel to display the **Data Types** panel, as shown in the following figure.

On the **Data Types** panel, you can reverse engineer Db2 objects.



```
ADB21E in                     DD1A Data Types               Row 1 to 11 of 925
Command ===>                                                  Scroll ===> PAGE
                                                                     More: >
Commands: GRANT
Line commands:
 T - Tables  A - Auth  AH - Schema auth  GR - Grant  DROP - Drop  COM - Comment
 I - Interpret  CRE - Create data type  GEN - Generate DDL  DDL - Object DDL
 ? - Show all line commands

                                Source                    M
S    Schema   Name              Schema   SourceType       T      Length  Scale
     *        *                 *        *                *           *      *
---- -------- -----------------> -------- ---------------> - ----------- ------
     NEWTON   FLOAT_ARRAY_123    SYSIBM   DOUBLE           A           8      0
     SCADI101 US_DOLLAR          SYSIBM   DECIMAL          T          15      2
     ADMF001  TYPE_TIMESTAMP_WIT SYSIBM   TIMESTAMP WITH T A          15     12
     ADMF001  MY_TYPE17          SYSIBM   INTEGER          A           4      0
     ADMF001  MY_BLOB1           SYSIBM   BLOB             A         500      0
     ADMF001  MY_TIME1_TIMESTAMP SYSIBM   TIMESTAMP        A          10      6
     ADMF001  TYPE_TIMESTAMP_WIT SYSIBM   TIMESTAMP WITH T A          15     12
     ADMF001  MY_TYPE1           SYSIBM   INTEGER          A           4      0
     ADMF001  MY_TIME1           SYSIBM   TIME             A           3      0
     ADMF001  MY_TYPE1_DATE      SYSIBM   DATE             A           4      0
     ADMF001  MY_TYPE1_CLOB      SYSIBM   CLOB             A     1048576      0
```

The following primary command is valid on this panel:

GRANT
Issues a GRANT command on multiple user-defined data types.

Recommendation: The GRANT command operates on each row that is displayed in the table. If you want to omit some of the rows before you issue the command, you can use the minus (-) line command to remove rows from the display. The GRANT command operates only on rows that are listed.

The following columns are on this panel:

S
Input field where you enter one of the line commands listed on the panel.

Schema
Schema of the data type.

Data Type Name
Name of the data type.

Source Schema
Schema of the source data type.

Source Data Type
Name of the source data type for this distinct data type.

MT
Metatype. Specify one of the following values:

T
Distinct.

A
Array.

L
Associative array.

Length
Maximum length for the data type, or precision for distinct types.

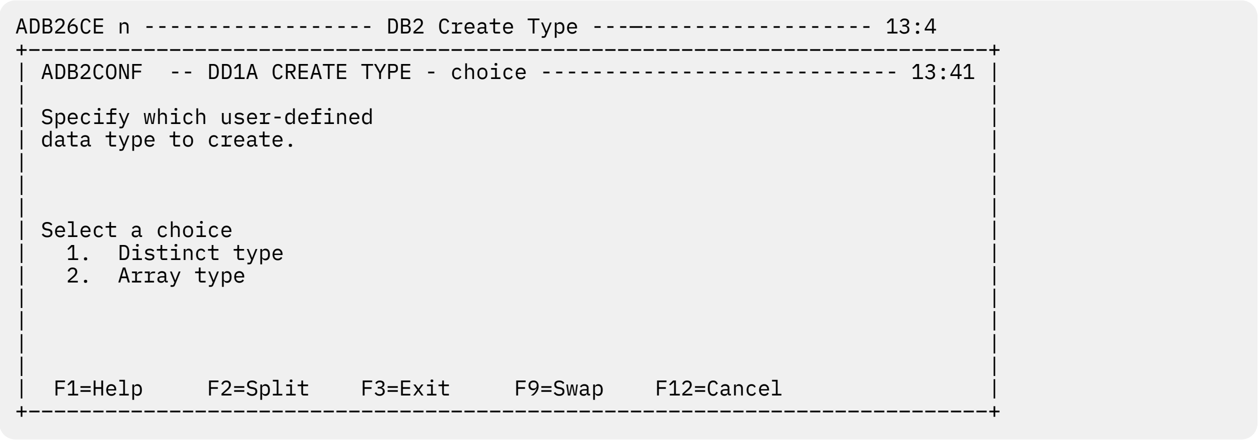
Scale
Scale for distinct data types, based on the built in decimal type.

Creating an array data type

Use the **Create Array Type** panel to create a new array type.

Procedure

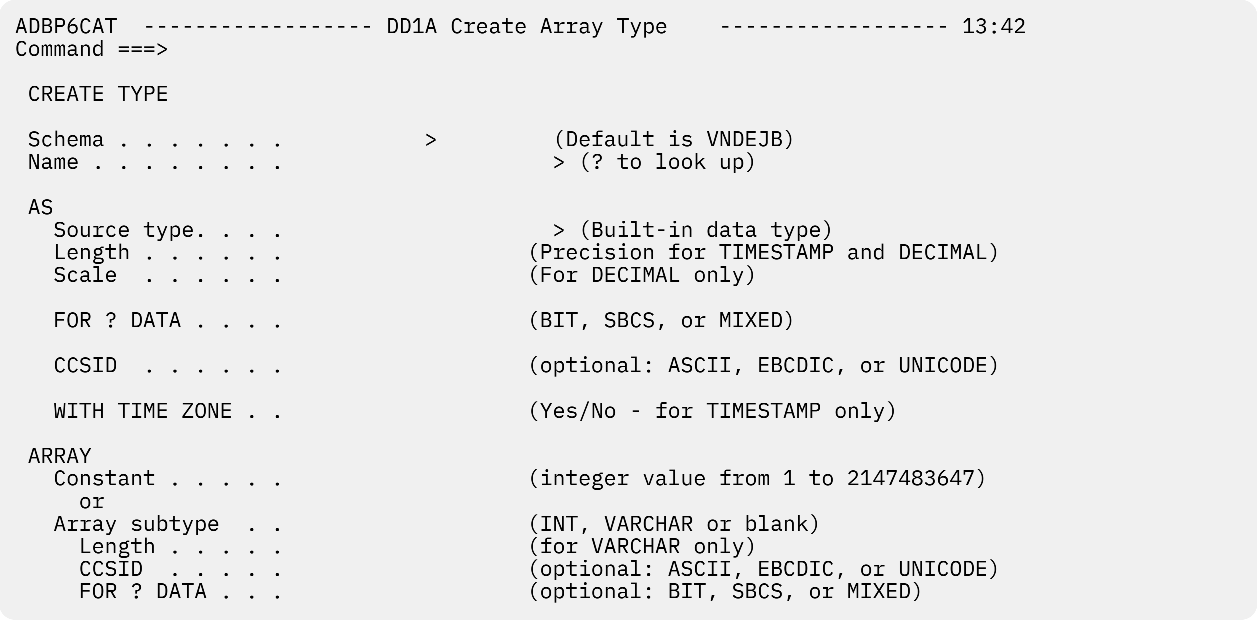
1. Specify option CE on the **Create/Drop/Label/Comment On Objects** panel. The **Create Type** panel is displayed, as shown in the following figure.



```
ADB26CE n ----------------- DB2 Create Type ---------------------- 13:4
+---------------------------------------------------------------------------+
| ADB2CONF  -- DD1A CREATE TYPE - choice ---------------------------- 13:41 |
|                                                                           |
| Specify which user-defined                                                |
| data type to create.                                                      |
|                                                                           |
|                                                                           |
|                                                                           |
| Select a choice                                                           |
|   1.  Distinct type                                                       |
|   2.  Array type                                                          |
|                                                                           |
|                                                                           |
|                                                                           |
|                                                                           |
|  F1=Help     F2=Split    F3=Exit     F9=Swap    F12=Cancel                |
+---------------------------------------------------------------------------+
```

Figure 32. Create Type panel

2. Specify option 2 for ARRAY TYPE. The **Create Array Type** panel is displayed, as shown in the following figure.



```
ADBP6CAT  ----------------- DD1A Create Array Type    ----------------- 13:42
Command ===>

 CREATE TYPE

 Schema . . . . . . .           >         (Default is VNDEJB)
 Name . . . . . . . .                     > (? to look up)

 AS
   Source type. . . .                     > (Built-in data type)
   Length . . . . . .                   (Precision for TIMESTAMP and DECIMAL)
   Scale  . . . . . .                   (For DECIMAL only)

   FOR ? DATA . . . .                   (BIT, SBCS, or MIXED)

   CCSID  . . . . . .                   (optional: ASCII, EBCDIC, or UNICODE)

   WITH TIME ZONE . .                   (Yes/No - for TIMESTAMP only)

 ARRAY
   Constant . . . . .                   (integer value from 1 to 2147483647)
     or
   Array subtype  . .                   (INT, VARCHAR or blank)
     Length . . . . .                   (for VARCHAR only)
     CCSID  . . . . .                   (optional: ASCII, EBCDIC, or UNICODE)
     FOR ? DATA . . .                   (optional: BIT, SBCS, or MIXED)
```

Figure 33. Create Array Type panel (ADBP6CAT)

3. Specify the following values for the array type:
   a) In the **Schema** field, enter the schema.
   b) In the **Name** field, enter the name.
   c) In the fields within the **AS** area, enter the information that goes inside the brackets of an AS clause.
      - In the **Source type** field, enter the name of the built-in data type.
      - If specifying a TIMESTAMP or DECIMAL, enter the length in the **Length** field.
      - If specifying a DECIMAL, enter the scale in the **Scale** field.
      - In the **FOR / DATA** field, BIT, SBCS, or MIXED.

- In the optional **CCSID** field, ASCII, EBCDIC, or UNICOD.
- If specifying a TIMESTAMP, enter YES or NO in the **WITH TIME ZONE** field.

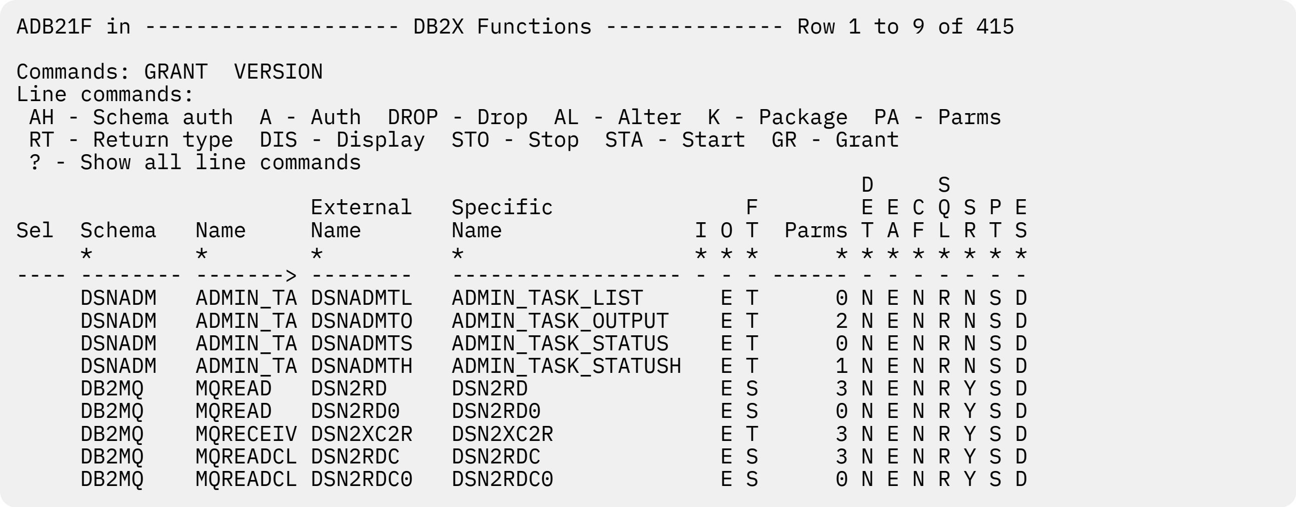
d) In the fields within the **ARRAY** area, enter the following fields. Array subtype and Constant are mutually exclusive. An error message is returned if both array subtype and constant are non-blank. Leave the fields blank if you want to use the Constant default value of 2147483647.

- In the **Array** subtype field, enter INT or VARCHAR.
- In the **Constant** field, enter an integer value from rom 1 to 2147483647..
- If specifying a varchar,, enter the length in the **Length** field.
- If specifying a varchar, optionally enter ASCII, EBCDIC, or UNICODE in the **CCSID** field.
- If specifying a varchar, optionally enter BIT, SBCS, or MIXED in the **FOR ? DATA** field.

Option F. Functions

Use the **Functions** panel to display information about the functions in the Db2 catalog.

Select option F on the **System Catalog** panel to display the **Functions** panel, as shown in the following figure.



```
ADB21F in -------------------- DB2X Functions -------------- Row 1 to 9 of 415

Commands: GRANT  VERSION
Line commands:
 AH - Schema auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
 RT - Return type  DIS - Display  STO - Stop  STA - Start  GR - Grant
 ? - Show all line commands
                                                                  D     S
                       External   Specific               F        E E C Q S P E
Sel  Schema   Name     Name       Name               I O T  Parms T A F L R T S
     *        *        *          *                  * * *      * * * * * * * *
---- -------- -------> --------   ------------------ - - - ------ - - - - - - -
     DSNADM   ADMIN_TA DSNADMTL   ADMIN_TASK_LIST      E T      0 N E N R N S D
     DSNADM   ADMIN_TA DSNADMTO   ADMIN_TASK_OUTPUT    E T      2 N E N R N S D
     DSNADM   ADMIN_TA DSNADMTS   ADMIN_TASK_STATUS    E T      0 N E N R N S D
     DSNADM   ADMIN_TA DSNADMTH   ADMIN_TASK_STATUSH   E T      1 N E N R N S D
     DB2MQ    MQREAD   DSN2RD     DSN2RD               E S      3 N E N R Y S D
     DB2MQ    MQREAD   DSN2RD0    DSN2RD0              E S      0 N E N R Y S D
     DB2MQ    MQRECEIV DSN2XC2R   DSN2XC2R             E T      3 N E N R Y S D
     DB2MQ    MQREADCL DSN2RDC    DSN2RDC              E S      3 N E N R Y S D
     DB2MQ    MQREADCL DSN2RDC0   DSN2RDC0             E S      0 N E N R Y S D
```

Figure 34. Functions panel (ADB21F)

The following primary command is valid on this panel:

GRANT
Issues a GRANT command on multiple functions.

Recommendation: The GRANT command operates on each row that is displayed in the table. If you want to omit some of the rows before you issue the command, you can use the minus (-) line command to remove rows from the display. The GRANT command operates only on rows that are listed.

The following columns are on this panel:

Sel
Input field where you enter one of the line commands listed on the panel.

Schema
Schema of the function.

Name
Name of the function.

External Name
Load module name for the stored procedure. This field is blank if it is not an external or user-defined function.

Specific Name
The specific name of the function.

I
Indicates if the routine is an inline function. Indicate Yes or No.

O
Origin of the function, which is one of the following values:

- **E**
  External.
- **U**
  Sourced.
- **S**
  System generated.
- **Q**
  SQL.

FT
Function type, which is one of the following types:

- **C**
  Column.
- **S**
  Scaler.
- **T**
  Table.

Parms
Number of parameters for the function.

DET
This field indicates whether the external function is deterministic (that is, returns the same result when called using the same parameters). This field contains one of the following values:

- **Y**
  Yes.
- **N**
  No.
- **blank**
  The routine is a function, but not an external function.

EA
This field indicates whether the external function changes the state of an object that Db2 does not manage. This field contains one of the following values:

- **E**
  Yes.
- **N**
  No.
- **blank**
  The routine is a stored procedure.

CF
Cast function, which is one of the following values:

- **Y**
  Yes.
- **N**
  No.

SQL

This field indicates whether SQL statements are allowed, which is one of the following values:

N

Contains no SQL statements.

C

Contains SQL statements.

R

Reads SQL data.

M

Modifies SQL data.

blank

Not applicable.

SR

This field indicates whether the program should remain resident when it ends. This field contains one of the following values:

Y

Program remains resident.

N

Program does not remain resident.

blank

Not external or user-defined function.

PT

Program type, which is one of the following types:

M

Main.

S

Subroutine.

blank

Not external or user-defined function.

ES

External security, which is one of the following values:

D

Db2 address space user.

U

User.

C

Definer.

blank

Not external or user-defined function.

Option G. Storage Groups

The **Storage Groups** panel displays the storage groups in the Db2 catalog.

Select option G on the **System Catalog** panel to display the **Storage Groups** panel, as shown in the following figure.

```
ADB21G in ---------------- DB2X Storage Groups ------------ Row 1 to 10 of 26

Line commands:
 D - Databases  S - Table spaces  X - Indexes  VOL - Volumes  I - Interpret
 GR - Grant  DROP - Drop  CRE - Create  AL - Alter  UT - Utility  A - Auth
 ? - Show all line commands

Select Name     Owner    VCAT           Space Statistics time
       *        *        *                  * *
------ -------- -------- -------- ----------- --------------------------
       ADBGCH   ADB      DB2X               0 0001-01-01-00.00.00.000000
       DSN8G81U DSCGDB2  DB2X               0 0001-01-01-00.00.00.000000
       DSN8G810 DSCGDB2  DB2X               0 0001-01-01-00.00.00.000000
       ISTJEG   ISTJE    DB2X               0 0001-01-01-00.00.00.000000
       SYSDEFLT DSCGDB2  DB2X               0 0001-01-01-00.00.00.000000
       TFLSG    ISTFL2   DB2X               0 0001-01-01-00.00.00.000000
```

Figure 35. Storage Groups panel (ADB21G)

The fields on this panel are:

Select
Input field where you enter one of the line commands listed on the panel.

Name
Name of the storage group.

Owner
Authorization ID of the owner of the storage group.

VCAT
Name of the VSAM or ICF catalog.

Space
Kilobytes (KB) of storage allocated for the storage group as determined by the STOSPACE utility the last time it was run. A value of -1 indicates that the utility has never been run.

Statistics Time
The timestamp of when the Space field was last updated.

Option GV. Global Variables

Use the **Global Variables** panel to display information about the global variables in the Db2 catalog.

Select option GV on the **System Catalog** panel to display the **Global Variables** panel, as shown in the following figure.

```
ADBP1GV n ---------------- DD1A Global Variables ---------- Row 1 to 11 of 325

Line commands:
 I - Interpretation  A - Auth  GEN - Generate DDL  DDL - Object DDL
 CRE - Create  COM - Comment  ALT - Alter  DROP - Drop  DO - Dependent objects
 ? - Show all line commands
                                   Data     Max
Select Schema   Name               Type     Length Scale  Default Text
       *        *                  *             *      * *
------ -------> ------------------ -------- <----- ------ ---------------------
       SYSIBM   CLIENT_IPADDR      CHAR         39      0 NULL
       SYSIBMAD GET_ARCHIVE        CHAR          1      0 'N'
       SYSIBMAD MOVE_TO_ARCHIVE    CHAR          1      0 'N'
       VNDRG    VAR1               INTEGER       4      0
       VNDRG    VAR2               VARCHAR     100      0
       VNDRG    VWINT              INTEGER       4      0
       VNDRG    TEXT               VARCHAR     128      0
       VNDRG    VARCHAR128         VARCHAR     128      0
       GVAR     TEST               VARCHAR     128      0
       GVAR     TESTFUNC           VARCHAR     128      0
       VNDRG    GINT               INTEGER       4      0
```

Figure 36. Global Variables panel (ADBP1GV)

The following primary commands are valid on this panel:

I
Interpretation. Provides detailed information about a specific global variable.

A
Authorization. display information about the users who grant privileges to global variables, and information about the users who hold the privileges.

GEN
Generate DDL. Generate SQL statements.

DDL
Object DDL.

CRE
Create.

COM
Comment. Object DDL.

ALT
Alter. Object DDL.

DROP
Drop. Object DDL.

DO
Dependent objects. Object DDL.

The following columns are on this panel:

Select
Input field where you enter one of the line commands listed on the panel.

Schema
The schema of the global variable.

Name
The name of the global variable.

Data Type
The name of the data type.

Max Length
The maximum length of the global variable.

Scale
The scale of the global variable.

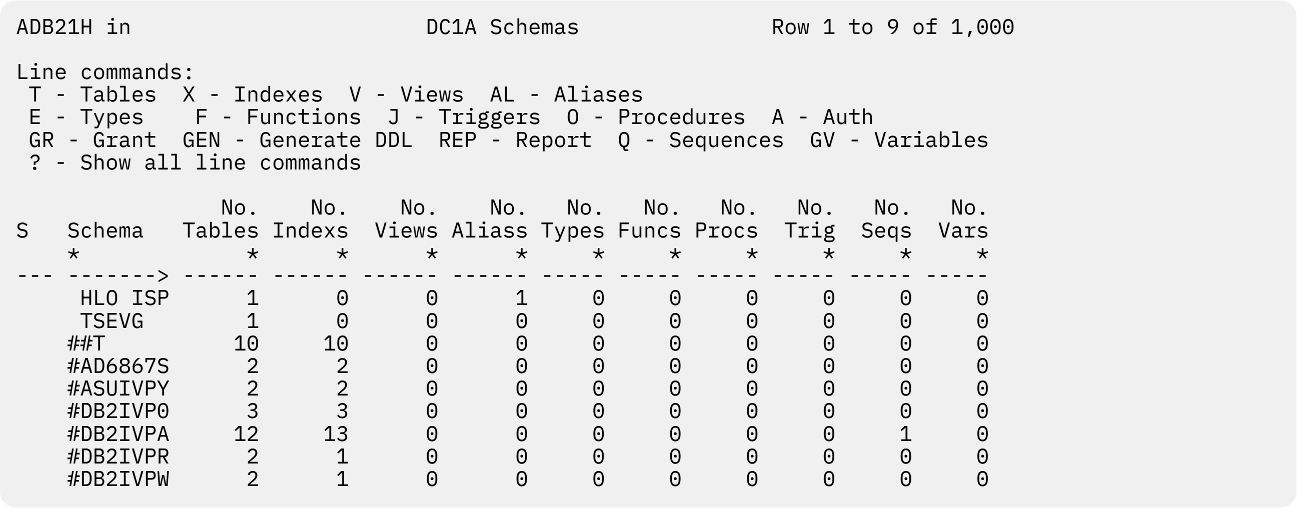
Default Text
The text of the default value of the global variable.

If the text is truncated, type EXPAND on the primary command line, position the cursor on the default text field, and press Enter to display all of the text.

Option H. Schemas

The **Schemas (ADB21H)** panel displays the schemas in the Db2 catalog.

To display the **Schemas (ADB21H)** panel, on the **System Catalog (ADB21)** panel, select option H.



```
ADB21H in                       DC1A Schemas               Row 1 to 9 of 1,000

Line commands:
 T - Tables  X - Indexes  V - Views  AL - Aliases
 E - Types    F - Functions  J - Triggers  O - Procedures  A - Auth
 GR - Grant  GEN - Generate DDL  REP - Report  Q - Sequences  GV - Variables
 ? - Show all line commands

                No.    No.    No.    No.   No.   No.   No.   No.   No.   No.
S   Schema   Tables Indexs  Views Aliass Types Funcs Procs  Trig  Seqs  Vars
    *             *      *      *      *     *     *     *     *     *     *
--- -------> ------ ------ ------ ------ ----- ----- ----- ----- ----- -----
     HLO ISP      1      0      0      1     0     0     0     0     0     0
     TSEVG        1      0      0      0     0     0     0     0     0     0
    ##T          10     10      0      0     0     0     0     0     0     0
    #AD6867S      2      2      0      0     0     0     0     0     0     0
    #ASUIVPY      2      2      0      0     0     0     0     0     0     0
    #DB2IVP0      3      3      0      0     0     0     0     0     0     0
    #DB2IVPA     12     13      0      0     0     0     0     0     1     0
    #DB2IVPR      2      1      0      0     0     0     0     0     0     0
    #DB2IVPW      2      1      0      0     0     0     0     0     0     0
```

Figure 37. ***Schemas (ADB21H)*** *panel*

Panel columns

This panel includes the following columns:

S
An input field where you can enter one of the line commands that are listed on the panel.

Schema
The schema of the objects.

No. Tables
The number of tables that are defined in this schema.

No. Indexs
The number of indexes that are defined in this schema.

No. Views
The number of views that are defined in this schema.

No. Aliass
The number of aliases that are defined in this schema.

No. Types
The number of distinct data types that are defined in this schema.

No. Funcs
The number of user-defined functions and implicitly-defined functions that are defined in this schema.

No. Procs
The number of stored procedures that are defined in this schema.

No. Trig
The number of table triggers that are defined in this schema.

No. Seqs
The number of sequences that are defined in this schema.

No. Vars
The number of variables that are defined in this schema.

Panel line commands

The following line commands are valid on this panel:

T
Display the tables in the schema.

X
Display the indexes in the schema.

V
Display the views in the schema.

AL
Display the aliases in the schema.

E
Display the data types in the schema.

F
Display the functions in the schema.

J
Display the triggers in the schema.

O
Display the procedures in the schema.

A
Display schema authorizations.

GR
Grant privileges on the schema.

GEN
Generate DDL for the schema. The number of objects processed by GEN might not match the number of objects displayed. For example, functions with CAST_FUNCTION=Y are not processed by GEN.

REP
Generate a report from the Db2 catalog.

Q
Display the sequences in the schema.

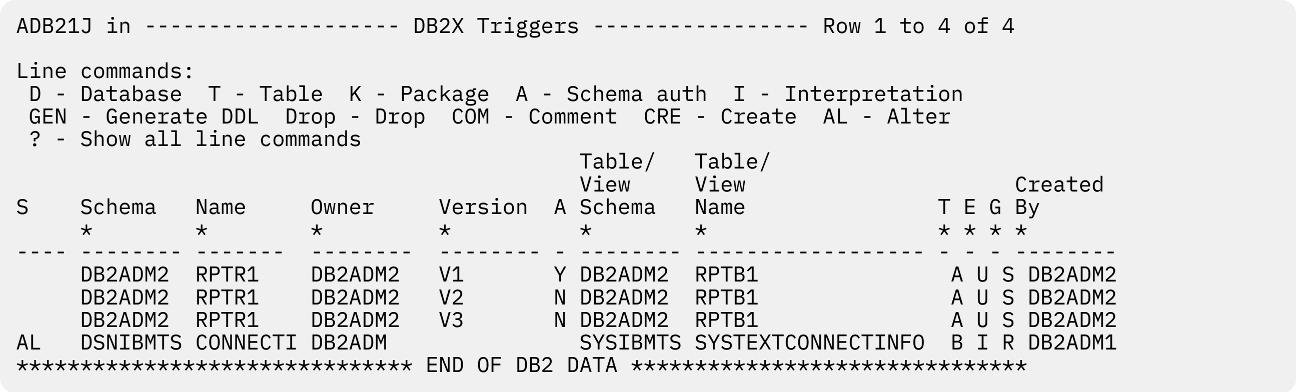
GV
Display the global variables in the schema.

CP
Set copy privileges.

Option J. Triggers

Use the **Triggers** panel to display information about the triggers in the Db2 catalog.

Select option J on the **System Catalog** pane to display the **Triggers** panel, as shown in the following figure.



```
ADB21J in -------------------- DB2X Triggers ----------------- Row 1 to 4 of 4

Line commands:
 D - Database  T - Table  K - Package  A - Schema auth  I - Interpretation
 GEN - Generate DDL  Drop - Drop  COM - Comment  CRE - Create  AL - Alter
 ? - Show all line commands
                                            Table/   Table/
                                            View     View                     Created
S    Schema   Name     Owner     Version   A Schema   Name               T E G By
     *        *        *         *          *        *                  * * * *
---- -------- -------  --------  -------- - -------- ------------------ - - - --------
     DB2ADM2  RPTR1    DB2ADM2   V1       Y DB2ADM2  RPTB1               A U S DB2ADM2
     DB2ADM2  RPTR1    DB2ADM2   V2       N DB2ADM2  RPTB1               A U S DB2ADM2
     DB2ADM2  RPTR1    DB2ADM2   V3       N DB2ADM2  RPTB1               A U S DB2ADM2
AL   DSNIBMTS CONNECTI DB2ADM               SYSIBMTS SYSTEXTCONNECTINFO  B I R DB2ADM1
******************************* END OF DB2 DATA *******************************
```

Figure 38. Triggers panel (ADB21J)

The following columns are on this panel:

S
Input field where you enter one of the line commands listed on the panel.

Schema
Name of the schema.

Name
Name of the trigger.

Owner
Authorization ID of the owner of the trigger.

Version
Trigger version for an advanced trigger.

A
Whether the trigger version is active for an advanced trigger:

- **Y**
  Yes.
- **N**
  No.
- **blank**
  The trigger is a basic trigger.

Table/View Schema
Schema of the table or view to which this trigger applies.

Table/View Name
Name of the table or view to which this trigger applies.

T
Trigger time, which is one of the following values:

- **A**
  After.
- **B**
  Before.
- **I**
  Instead of.

E
Trigger event, which is one of the following values:

- **I**
  Insert.
- **U**
  Update.
- **D**
  Delete.

G
Granularity of the trigger, which is one of the following values:

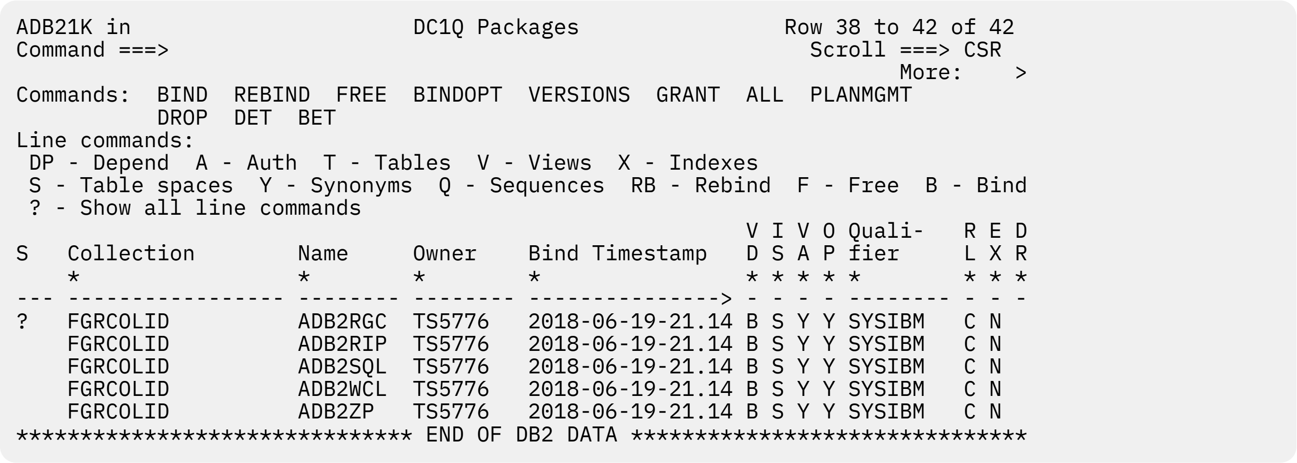
- **R**
  For each row.
- **S**
  For each statement.

Created By
Primary authorization ID of the user who created the trigger.

Option K. Packages

The **Packages (ADB21K)** panel displays the packages in the Db2 catalog.

To display the **Packages (ADB21K)** panel, select option K on the **System Catalog (ADB21)** panel.



```
ADB21K in                      DC1Q Packages                Row 38 to 42 of 42
Command ===>                                                  Scroll ===> CSR
                                                                     More:    >
Commands:  BIND  REBIND  FREE  BINDOPT  VERSIONS  GRANT  ALL  PLANMGMT
           DROP  DET  BET
Line commands:
 DP - Depend  A - Auth  T - Tables  V - Views  X - Indexes
 S - Table spaces  Y - Synonyms  Q - Sequences  RB - Rebind  F - Free  B - Bind
 ? - Show all line commands
                                                         V I V O Quali-   R E D
S   Collection        Name     Owner    Bind Timestamp   D S A P fier     L X R
    *                 *        *        *                * * * * *        * * *
--- ----------------- -------- -------- ---------------> - - - - -------- - - -
?   FGRCOLID          ADB2RGC  TS5776   2018-06-19-21.14 B S Y Y SYSIBM   C N
    FGRCOLID          ADB2RIP  TS5776   2018-06-19-21.14 B S Y Y SYSIBM   C N
    FGRCOLID          ADB2SQL  TS5776   2018-06-19-21.14 B S Y Y SYSIBM   C N
    FGRCOLID          ADB2WCL  TS5776   2018-06-19-21.14 B S Y Y SYSIBM   C N
    FGRCOLID          ADB2ZP   TS5776   2018-06-19-21.14 B S Y Y SYSIBM   C N
******************************* END OF DB2 DATA *******************************
```

Notes:

1. Panel ADB21K can also have the title **Packages Copy**. This alternate title is used if the panel displays information about copies of the package. [To display information about package copies, specify the KC line command on the **Packages (ADB21K)** panel.]
2. Panel ADB21K can also include the applicable plan name in the title (Packages for Plan *plan_name*). The plan name is included if this panel is displayed after specifying K on the **Application Plans (ADB21P)** panel or **Package List (ADB21PL)** panel.

*Figure 39. **Packages (ADB21K)** panel*

Panel columns

This panel includes the following columns:

S
An input field where you can enter one of the line commands that are listed on the panel.

Collection
The name of the package collection.

Name
The name of the package.

Owner
The authorization ID of the package owner.

Version
The version of the package.

Alternatively, one of the following columns might be displayed instead of **Version**: (The value that is displayed is controlled by the "VERSIONS" on page 158 command.)

Con Token
The consistency token for the package.

Bind Timestamp
The timestamp of when the package was last bound.

Depending on the value that was specified with the VERSIONS command, the version and consistency token might be displayed below the bind timestamp.

VD

An indication of whether validity checking can be deferred until run time. This field contains one of the following values:

B

All validity checking must be done during the bind.

R

Validity checking is done at run time for tables, views, and privileges that do not exist at bind time.

IS

The isolation level, which can be one of the following values:

R

Repeatable read.

S

Cursor stability.

T

Read stability.

U

Uncommitted read.

Blank

Not specified; therefore, the isolation level of the plan is used.

VA

An indication of whether the package is valid. A *valid plan* can be run without being rebound. This field contains one of the following values:

Y

Yes.

N

No.

A

An ALTER statement changed the description of the table or base table of a view, but the package is still valid.

H

An ALTER TABLE statement changed the description of the table or base table of a view. For releases of Db2 prior to Version 5, the change invalidates the package.

OP

An indication of whether the package can be allocated. This field contains one of the following values:

Y

Yes.

N

No. An explicit bind or rebind operation is required before the package can be allocated.

Qualifier

The qualifier that was specified at bind time to resolve names.

RL

The time at which resources for the package are released. This field contains one of the following values:

C

Commit time.

D

Deallocation time.

Blank

Not specified; therefore, the value that is specified for the plan is used.

EX

An indication of whether the package was bound using EXPLAIN. This field contains one of the following values:

Y

Yes.

N

No.

Only

EXPLAIN was run, EXPLAIN tables were populated, and the BIND process completed; however, the existing package was not affected.

DR

The DYNAMICRULES option that was used when the package was last bound. Possible values are any valid value for the DYNAMICRULES column of the Db2 SYSIBM.SYSPACKAGE catalog table, as described in SYSPACKAGE catalog table (Db2 12 for z/OS).

You can scroll to view additional columns. Those columns display information from the SYSIBM.SYSPACKAGE catalog table. For information about those column values, see SYSPACKAGE catalog table (Db2 12 for z/OS).

Panel primary commands

Tip: The primary commands operate only on rows that are listed on the panel. To omit some of the rows before you issue one of these commands, use the minus (-) line command to remove rows from the display.

The following primary commands are valid on this panel:

BIND

Issues a BIND command on multiple packages.

Restriction: You cannot BIND a trigger package by using the **Packages (ADB21K)** panel.

BINDOPT

Lets you choose bind and rebind options that are not in the Db2 catalog.

REBIND

Issues a REBIND command on multiple packages.

When you specify REBIND, the resulting BIND command contains only the package name. If you want the resulting BIND command to contain the package name and all of the parameters, specify REBIND FULL.

FREE

Issues a FREE command on multiple packages.

Notes for BIND, REBIND, and FREE:

- When you attempt to bind, rebind, or free more than 20 packages, Db2 Admin Tool prompts you to specify either a work statement list or a batch job to complete the required processing.
- If you activated prompting (see "Changing Db2 Admin Tool prompt options" on page 243), the **Statement Execution Prompt (ADB2PSTM)** panel is displayed. Select option 1A to execute all statements. The stacked output is displayed after the commands are processed.

VERSIONS

Displays the package version, bind timestamp, and consistency token in the fifth column. You can issue one of the following variations of the VERSIONS command:

VER ON

Displays the bind timestamp, with the version and consistency token listed below it.

VER SHORT

Displays only the package version.

VER OFF
Displays only the bind timestamp.

VER CON
Displays only the consistency token.

VER
Cycles between the VER SHORT, VER ON, VER CON, and VER OFF displays each time that you issue this command.

GRANT
Issues a GRANT command on multiple application packages.

ALL T
Displays all tables for the listed packages.

ALL X
Displays all tables for the listed packages.

PLANMGMT
Displays the plan management attributes for the package. When the PLANMGMT command is issued, the panel includes the QUALIFIER command.

QUALIFIER
Displays the qualifier for the package. You must first issue the PLANMGMT command to use the QUALIFIER command.

DROP
Drops the selected packages.

DET
Displays the detail package report for the selected packages.

BET
Generates a detail report in batch for the selected packages.

Panel line commands

The following line commands are valid on this panel:

.
Selects multiple packages for the REBIND, BIND, FREE, ALL, GRANT, and DROP commands.

/
Displays all of the columns for the selected row.

A
Displays authorizations for the packages.

B
Binds the packages.

BC
Binds and copies the packages.

BET
Generates a detailed report in batch for the package.

D
Shows databases that are associated with the package.

DET
Generates a detailed report online for the package.

DP
Shows dependencies.

DRP
Drops the packages.

EN
Displays enabled and disabled connections for the packages.

F
Frees the packages.

GR
Grants package privileges.

I
Displays details about the packages.

KC
Displays previous or original copies of the package.

LP
Lists PLAN_TABLE rows for the last BIND with EXPLAIN(YES).

LPA
Lists all the PLAN_TABLE rows that are associated with the package.

O
Displays stored procedures that are associated with the package.

P
Displays application plans that are associated with the package.

PL
Displays the package list items.

Q
Displays sequences that are associated with the package.

RB
Rebinds the packages.

RD
Regenerate a DBRM member.

RO
Shows the owner roles.

RS
Shows the REST services associated with the package.

S
Shows that table spaces that are associated with the package.

SQ
Shows the SQL statements in the package.

T
Shows the tables that are associated with the package.

V
Shows the views that are associated with the package.

VE
Shows the version and consistency tokens.

X
Shows the indexes that are associated with the package.

Y
Shows the synonyms that are associated with the package.

Related tasks

“Binding packages and generating BIND statements” on page 916
If you changed the SQL in your application, you need to bind the associated package to replace the existing one. You might also want to generate BIND statements for existing packages and save those statements without executing them.

“Freeing packages” on page 919

When you free, or delete, a package, the corresponding package information is deleted from the Db2 catalog. You can delete a specific version of a package, all versions of a package, or whole collections of packages.

“Displaying detailed package information” on page 919
Db2 Admin Tool can report detailed information for one or more packages, including SQL and EXPLAIN information.

“Viewing SQL statements for a package” on page 922
You can use Db2 Admin Tool to view the SQL statements that are included in a package.

“ Regenerating DBRMs” on page 927
Db2 Admin Tool can regenerate any missing DBRMs for you based on the information in the Db2 catalog table SYSIBM.SYSPACKSTMT.

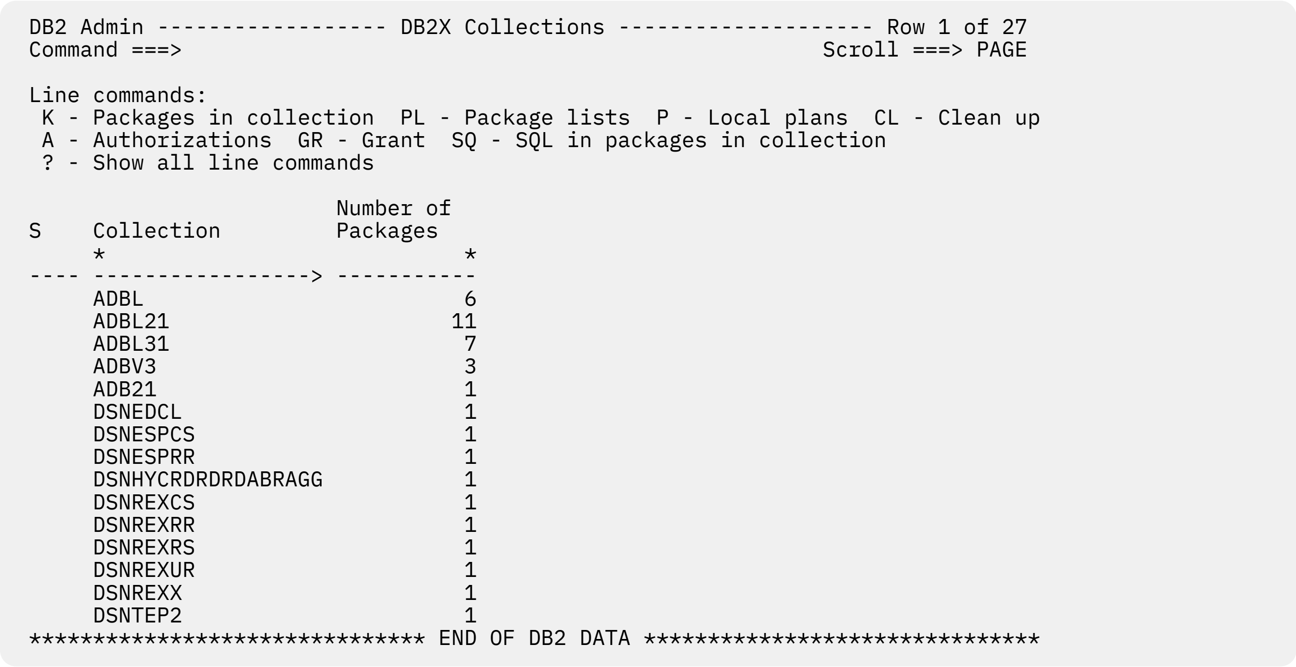
Related reference

“Option K. Packages” on page 156
The **Packages (ADB21K)** panel displays the packages in the Db2 catalog.

Option L. Collections

The **Collections (ADB21L)** panel displays the collections that are recorded in the Db2 catalog. A *collection* is a group of associated packages.

To display the **Collections (ADB21L)** panel, on the **System Catalog (ADB21)** panel, select option L.



```
DB2 Admin ------------------ DB2X Collections -------------------- Row 1 of 27
Command ===>                                                  Scroll ===> PAGE

Line commands:
 K - Packages in collection  PL - Package lists  P - Local plans  CL - Clean up
 A - Authorizations  GR - Grant  SQ - SQL in packages in collection
 ? - Show all line commands

                        Number of
S    Collection         Packages
     *                            *
---- -----------------> -----------
     ADBL                         6
     ADBL21                      11
     ADBL31                       7
     ADBV3                        3
     ADB21                        1
     DSNEDCL                      1
     DSNESPCS                     1
     DSNESPRR                     1
     DSNHYCRDRDRDABRAGG           1
     DSNREXCS                     1
     DSNREXRR                     1
     DSNREXRS                     1
     DSNREXUR                     1
     DSNREXX                      1
     DSNTEP2                      1
******************************* END OF DB2 DATA *******************************
```

*Figure 40. **Collections (ADB21L)** panel*

Panel columns

This panel includes the following columns:

S
An input field where you can enter one of the line commands that are listed on the panel.

Collection
The name of the package collection.

Number of Packages
The number of packages in the collection.

Related tasks

“Viewing SQL statements for a collection” on page 923
You can use Db2 Admin Tool to view the SQL statements that are included in a collection of packages.

“Deleting obsolete packages” on page 925
Deleting obsolete packages helps clean up your package collections and the Db2 catalog.

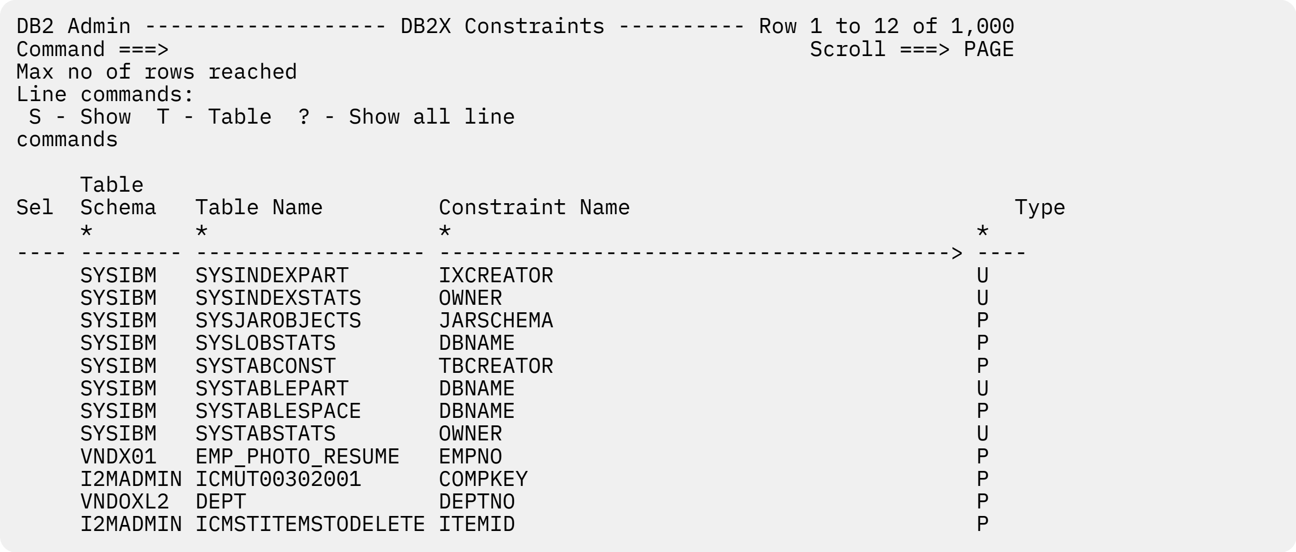
Related reference

Preparation process for an application program (Db2 12 for z/OS)

Option N. Constraints

The Constraints panel displays the constraints on a table in the Db2 catalog.

Select option N on the **System Catalog** panel to display the **Constraints** panel, as shown in the following figure.



```
DB2 Admin ------------------- DB2X Constraints ---------- Row 1 to 12 of 1,000
Command ===>                                                  Scroll ===> PAGE
Max no of rows reached
Line commands:
 S - Show  T - Table  ? - Show all line
commands

     Table
Sel  Schema   Table Name         Constraint Name                              Type
     *        *                  *                                         *
---- -------- ------------------ ----------------------------------------> ----
     SYSIBM   SYSINDEXPART       IXCREATOR                                 U
     SYSIBM   SYSINDEXSTATS      OWNER                                     U
     SYSIBM   SYSJAROBJECTS      JARSCHEMA                                 P
     SYSIBM   SYSLOBSTATS        DBNAME                                    P
     SYSIBM   SYSTABCONST        TBCREATOR                                 P
     SYSIBM   SYSTABLEPART       DBNAME                                    U
     SYSIBM   SYSTABLESPACE      DBNAME                                    P
     SYSIBM   SYSTABSTATS        OWNER                                     U
     VNDX01   EMP_PHOTO_RESUME   EMPNO                                     P
     I2MADMIN ICMUT00302001      COMPKEY                                   P
     VNDOXL2  DEPT               DEPTNO                                    P
     I2MADMIN ICMSTITEMSTODELETE ITEMID                                    P
```

Figure 41. Constraints panel (ADB21N) – partial display

The following columns are on this panel:

Sel
Enter one of the line commands listed on the panel.

Table Schema
The schema of the table on which the constraint is defined.

Table Name
The name of the table.

Constraint Name
The name of the constraint.

Type
The type of constraint. The following types are valid:

P
Primary key.

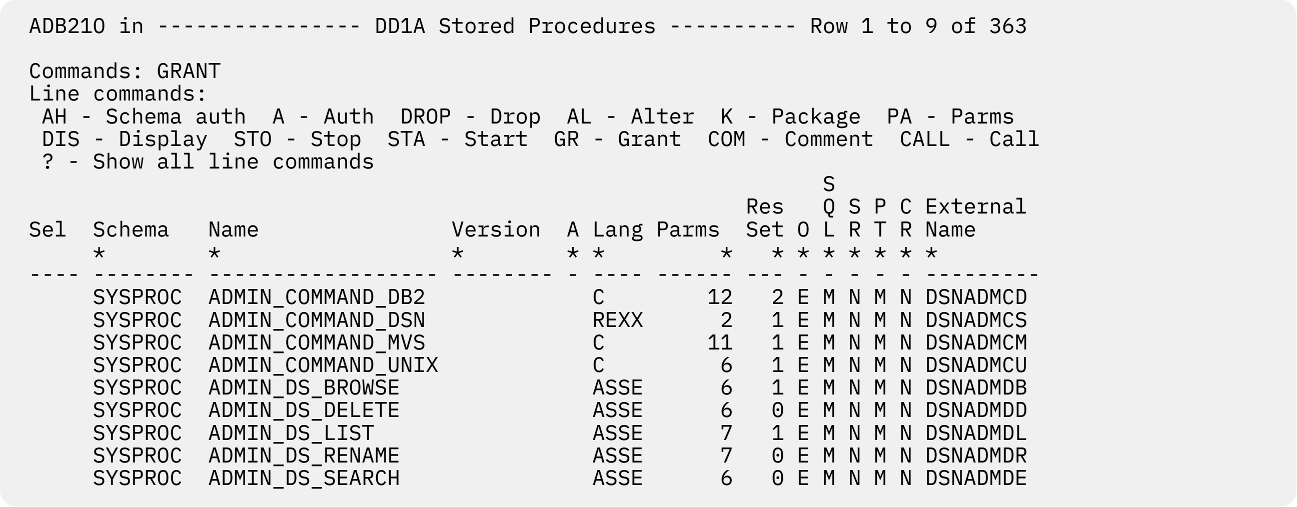
U
Unique.

F
Foreign key.

Option O. Stored Procedures

Use the **Stored Procedures** panel to display information about the stored procedures in the Db2 catalog.

Select option O on the **System Catalog** panel to display the **Stored Procedures** panel, as shown in the following figure.



```
ADB21O in ---------------- DD1A Stored Procedures ---------- Row 1 to 9 of 363

Commands: GRANT
Line commands:
 AH - Schema auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
 DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment  CALL - Call
 ? - Show all line commands
                                                              S
                                                        Res   Q S P C External
Sel  Schema   Name               Version  A Lang Parms  Set O L R T R Name
     *        *                  *        * *         *   * * * * * * *
---- -------- ------------------ -------- - ---- ------ --- - - - - - ---------
     SYSPROC  ADMIN_COMMAND_DB2             C        12   2 E M N M N DSNADMCD
     SYSPROC  ADMIN_COMMAND_DSN             REXX      2   1 E M N M N DSNADMCS
     SYSPROC  ADMIN_COMMAND_MVS             C        11   1 E M N M N DSNADMCM
     SYSPROC  ADMIN_COMMAND_UNIX            C         6   1 E M N M N DSNADMCU
     SYSPROC  ADMIN_DS_BROWSE               ASSE      6   1 E M N M N DSNADMDB
     SYSPROC  ADMIN_DS_DELETE               ASSE      6   0 E M N M N DSNADMDD
     SYSPROC  ADMIN_DS_LIST                 ASSE      7   1 E M N M N DSNADMDL
     SYSPROC  ADMIN_DS_RENAME               ASSE      7   0 E M N M N DSNADMDR
     SYSPROC  ADMIN_DS_SEARCH               ASSE      6   0 E M N M N DSNADMDE
```

Figure 42. Stored Procedures panel (ADB21O)

The following primary command is valid on this panel:

GRANT
Issues a GRANT command on multiple stored procedures.

Tip: The GRANT command operates on each row that is displayed in the table. If you want to omit some of the rows before you issue the command, you can use the minus (-) line command to remove rows from the display. The GRANT command operates only on rows that are listed.

The following columns are on this panel:

Sel
Input field where you enter one of the line commands listed on the panel.

Schema
Schema of the stored procedure.

Name
Name of the stored procedure.

Version
The version identifier for a native SQL procedure.

A
Identifies the active version of a routine.

Lang
Implementation language.

Parms
Number of parameters for the stored procedure.

Res Set
Maximum number of result sets that can be returned.

O
Origin of the routine:

E
External.

Q
SQL.

S
System generated.

U
User-defined or built-in function.

N
Native SQL procedure.

SQL
This field indicates whether SQL statements are allowed, which is one of the following values:

N
No SQL statement, SQL is not allowed.

C
Contains SQL statements.

R
Reads SQL data.

M
Modifies SQL data.

blank
Not applicable.

SR
This field indicates whether the program should remain resident when it ends. This field contains one of the following values:

Y
Program remains resident.

N
Program does not remain resident.

blank
Not external or user-defined function.

PT
Program type, which is one of the following values:

M
Main.

S
Subroutine.

blank
Not applicable.

CR
Commit on return. This field contains one of the following values:

Y
Unit of work is committed immediately.

N
Unit of work continues.

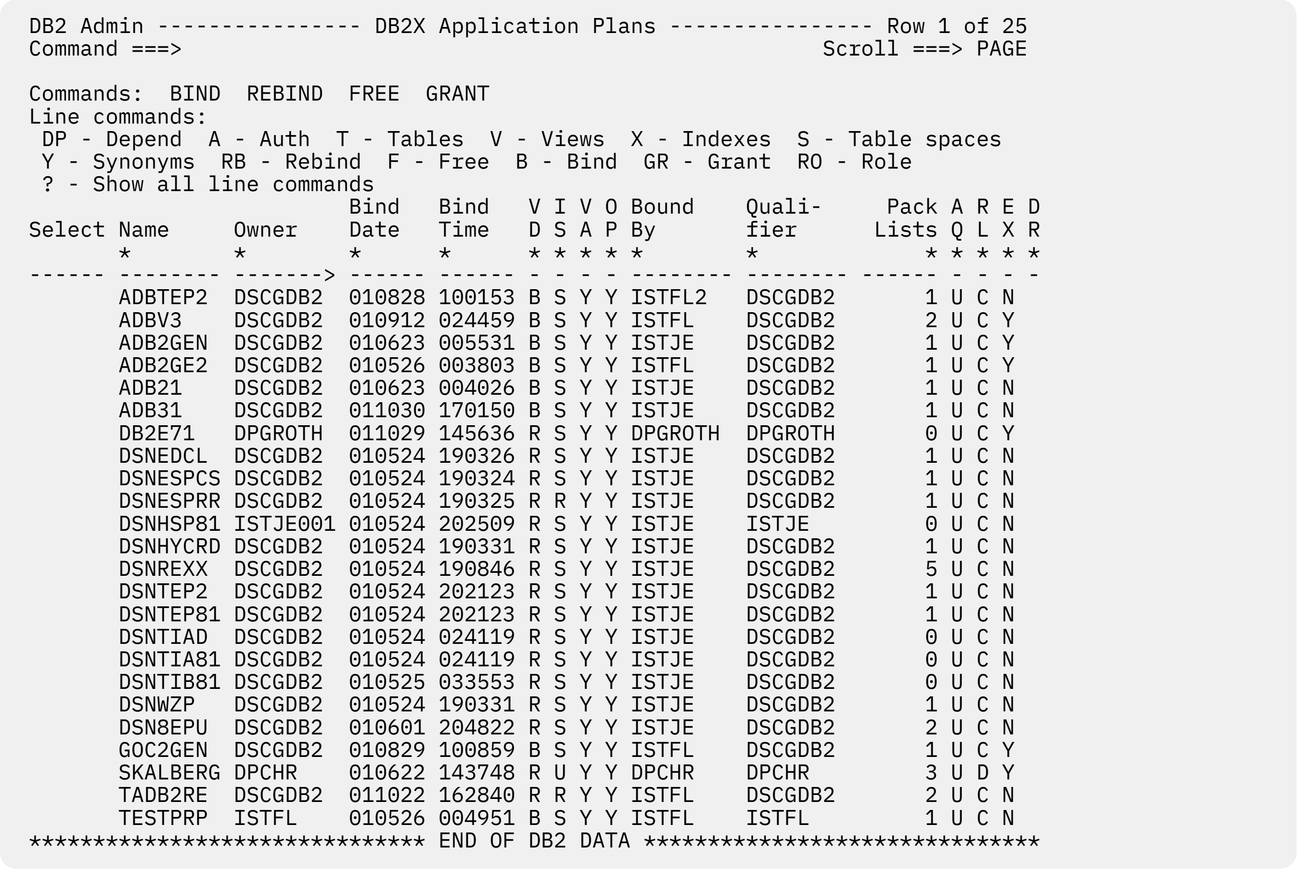
A
Autonomous. Only the unit of work from the procedure is committed. Work from the application that calls the procedure is not immediately committed.

External Name
Load module name for the stored procedure.

Option P. Plans

The **Application Plans (ADB21P)** panel displays the application plans in the Db2 catalog.

On the **System Catalog (ADB21)** panel, specify option P to display the **Application Plans (ADB21P)** panel:



```
DB2 Admin ---------------- DB2X Application Plans ---------------- Row 1 of 25
Command ===>                                                  Scroll ===> PAGE

Commands:  BIND  REBIND  FREE  GRANT
Line commands:
 DP - Depend  A - Auth  T - Tables  V - Views  X - Indexes  S - Table spaces
 Y - Synonyms  RB - Rebind  F - Free  B - Bind  GR - Grant  RO - Role
 ? - Show all line commands
                         Bind   Bind   V I V O Bound    Quali-      Pack A R E D
Select Name     Owner    Date   Time   D S A P By       fier       Lists Q L X R
       *        *        *      *      * * * * *        *              * * * * *
------ -------- -------> ------ ------ - - - - -------- -------- ------ - - - -
       ADBTEP2  DSCGDB2  010828 100153 B S Y Y ISTFL2   DSCGDB2       1 U C N
       ADBV3    DSCGDB2  010912 024459 B S Y Y ISTFL    DSCGDB2       2 U C Y
       ADB2GEN  DSCGDB2  010623 005531 B S Y Y ISTJE    DSCGDB2       1 U C Y
       ADB2GE2  DSCGDB2  010526 003803 B S Y Y ISTFL    DSCGDB2       1 U C Y
       ADB21    DSCGDB2  010623 004026 B S Y Y ISTJE    DSCGDB2       1 U C N
       ADB31    DSCGDB2  011030 170150 B S Y Y ISTJE    DSCGDB2       1 U C N
       DB2E71   DPGROTH  011029 145636 R S Y Y DPGROTH  DPGROTH       0 U C Y
       DSNEDCL  DSCGDB2  010524 190326 R S Y Y ISTJE    DSCGDB2       1 U C N
       DSNESPCS DSCGDB2  010524 190324 R S Y Y ISTJE    DSCGDB2       1 U C N
       DSNESPRR DSCGDB2  010524 190325 R R Y Y ISTJE    DSCGDB2       1 U C N
       DSNHSP81 ISTJE001 010524 202509 R S Y Y ISTJE    ISTJE         0 U C N
       DSNHYCRD DSCGDB2  010524 190331 R S Y Y ISTJE    DSCGDB2       1 U C N
       DSNREXX  DSCGDB2  010524 190846 R S Y Y ISTJE    DSCGDB2       5 U C N
       DSNTEP2  DSCGDB2  010524 202123 R S Y Y ISTJE    DSCGDB2       1 U C N
       DSNTEP81 DSCGDB2  010524 202123 R S Y Y ISTJE    DSCGDB2       1 U C N
       DSNTIAD  DSCGDB2  010524 024119 R S Y Y ISTJE    DSCGDB2       0 U C N
       DSNTIA81 DSCGDB2  010524 024119 R S Y Y ISTJE    DSCGDB2       0 U C N
       DSNTIB81 DSCGDB2  010525 033553 R S Y Y ISTJE    DSCGDB2       0 U C N
       DSNWZP   DSCGDB2  010524 190331 R S Y Y ISTJE    DSCGDB2       1 U C N
       DSN8EPU  DSCGDB2  010601 204822 R S Y Y ISTJE    DSCGDB2       2 U C N
       GOC2GEN  DSCGDB2  010829 100859 B S Y Y ISTFL    DSCGDB2       1 U C Y
       SKALBERG DPCHR    010622 143748 R U Y Y DPCHR    DPCHR         3 U D Y
       TADB2RE  DSCGDB2  011022 162840 R R Y Y ISTFL    DSCGDB2       2 U C N
       TESTPRP  ISTFL    010526 004951 B S Y Y ISTFL    ISTFL         1 U C N
******************************* END OF DB2 DATA *******************************
```

*Figure 43. **Application Plans (ADB21P)** panel*

On this panel, you can issue line commands to bind, rebind, and free application plans. You can also view the SQL statements by issuing the SQ line command. However, be aware that because the SQ line command applies to all packages in a plan, this command might affect performance.

Application Plans (ADB21P) panel has the following primary commands:

BIND
Issues a BIND command on multiple application plans.

FREE
Issues a FREE command on multiple application plans.

REBIND
Issues a REBIND command on multiple application plans.

When you specify REBIND, the resulting BIND command contains only the plan name. If you want the resulting BIND command to contain the plan name and all of the parameters, specify REBIND FULL.

Note: If you attempt to bind, free, or rebind more than 20 application plans, Db2 Admin Tool prompts you to specify either a work statement list or a batch job to complete the processing.

GRANT
Issues a GRANT command on multiple application plans.

Tip: The BIND, REBIND, FREE, and GRANT commands operate on each row that is displayed in the table. If you want to omit some of the rows before you issue one of these commands, you can use the minus (-)

line command to remove rows from the display. The BIND, REBIND, FREE, and GRANT commands operate only on rows that are listed.

Application Plans (ADB21P) panel has the following columns:

Select
Input field where you enter one of the line commands that are listed on the panel.

Name
Name of the application plan.

Owner
Authorization ID of the owner of the application plan.

Bind Date
Date of the most recent bind on the application plan. The date is in the form YYMMDD.

Bind Time
Time of the most recent bind on the application plan. The time is in the form HHMMSS.

VD
An indication of whether validity checking can be deferred until run time. This field contains one of the following values:

B
All validity checking must be done during the bind.

R
Validity checking is done at run time for tables, views, and privileges that do not exist at bind time.

IS
Isolation level, which is one of the following values:

R
Repeatable read.

S
Cursor stability.

T
Read stability.

U
Uncommitted read.

VA
An indication of whether the application plan is valid; that is, whether it can be run without being rebound. This field contains one of the following values:

Y
A valid application plan.

N
Not a valid application plan.

A
The description changed. The application plan is still valid.

H
The description changed. The application plan is valid for Db2 Version 5 or higher; otherwise, the plan is invalid.

OP
An indication of whether the application plan can be allocated. This field contains one of the following values:

Y
Yes.

N
No. An explicit bind or rebind is required before the plan can be allocated.

Bound By
Primary authorization ID of the binder of the plan.

Qualifier
Qualifier that was specified at bind time to resolve names.

Pack Lists
Number of packages in the package list at bind time.

AQ
Time at which resources for the application plan are acquired. This field contains one of the following values:

A
At allocation time

U
At first use

RL
Time at which resources for the application plan are released. This field contains one of the following values:

C
At commit time

D
At deallocation time

EX
An indication of whether the application plan was bound using EXPLAIN. This field contains one of the following values:

Y
Yes

N
No

DR
Dynamic SQL rules. This field contains one of the following values:

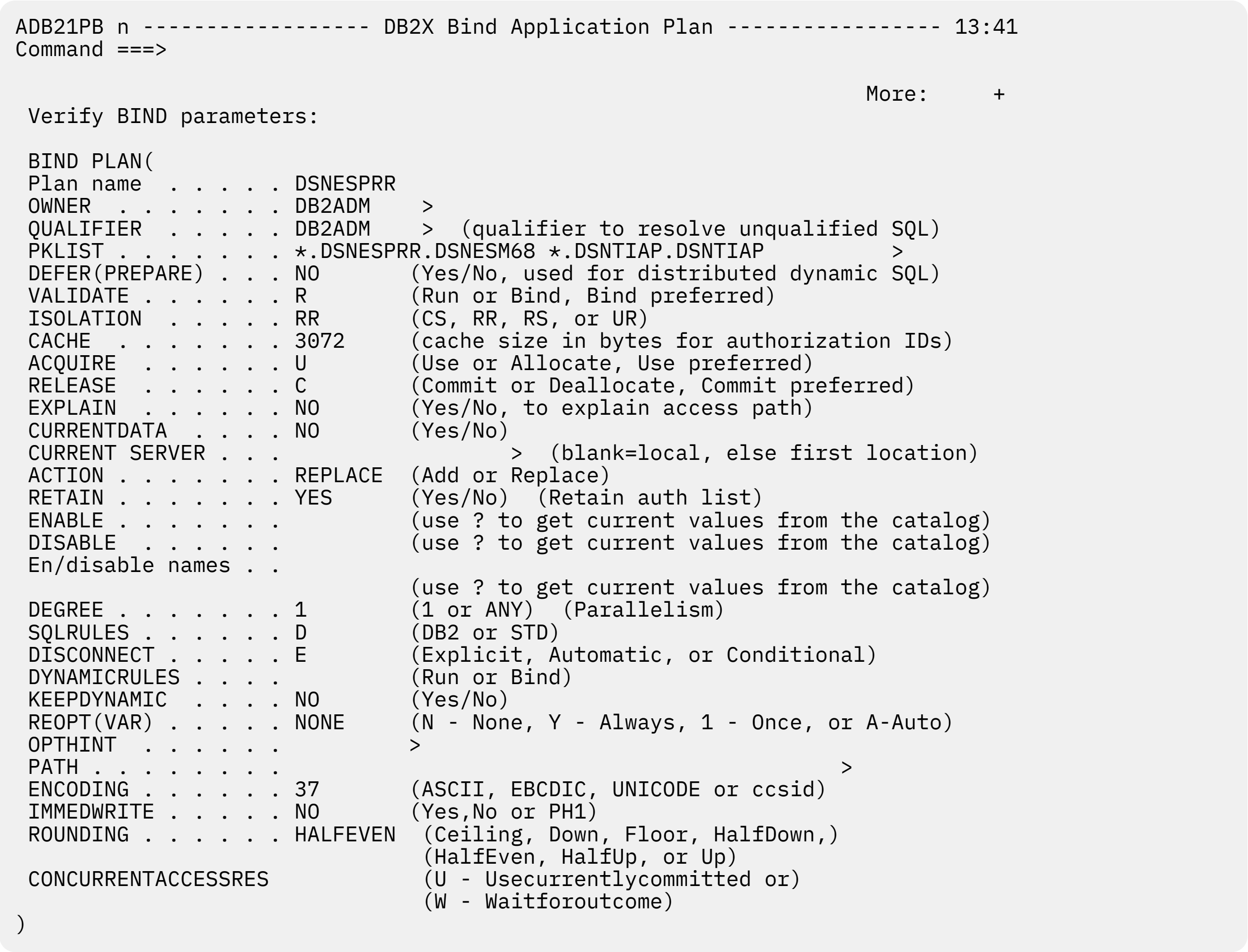
B
Use the AUTHID and authorizations of the binder

Blank
Use the AUTHID and authorizations of the executor

Binding application plans

To bind an application plan:

1. On the **Application Plans (ADB21P)** panel, specify the B line command (bind plan) and press Enter.
2. On the **Bind Application Plan (ADB21PB)** panel, enter your input and press Enter:



```
ADB21PB n ------------------ DB2X Bind Application Plan ----------------- 13:41
Command ===>

                                                                   More:     +
 Verify BIND parameters:

 BIND PLAN(
 Plan name  . . . . . DSNESPRR
 OWNER  . . . . . . . DB2ADM    >
 QUALIFIER  . . . . . DB2ADM    >  (qualifier to resolve unqualified SQL)
 PKLIST . . . . . . . *.DSNESPRR.DSNESM68 *.DSNTIAP.DSNTIAP          >
 DEFER(PREPARE) . . . NO       (Yes/No, used for distributed dynamic SQL)
 VALIDATE . . . . . . R        (Run or Bind, Bind preferred)
 ISOLATION  . . . . . RR       (CS, RR, RS, or UR)
 CACHE  . . . . . . . 3072     (cache size in bytes for authorization IDs)
 ACQUIRE  . . . . . . U        (Use or Allocate, Use preferred)
 RELEASE  . . . . . . C        (Commit or Deallocate, Commit preferred)
 EXPLAIN  . . . . . . NO       (Yes/No, to explain access path)
 CURRENTDATA  . . . . NO       (Yes/No)
 CURRENT SERVER . . .                  >  (blank=local, else first location)
 ACTION . . . . . . . REPLACE  (Add or Replace)
 RETAIN . . . . . . . YES      (Yes/No)  (Retain auth list)
 ENABLE . . . . . . .          (use ? to get current values from the catalog)
 DISABLE  . . . . . .          (use ? to get current values from the catalog)
 En/disable names . .
                               (use ? to get current values from the catalog)
 DEGREE . . . . . . . 1        (1 or ANY)  (Parallelism)
 SQLRULES . . . . . . D        (DB2 or STD)
 DISCONNECT . . . . . E        (Explicit, Automatic, or Conditional)
 DYNAMICRULES . . . .          (Run or Bind)
 KEEPDYNAMIC  . . . . NO       (Yes/No)
 REOPT(VAR) . . . . . NONE     (N - None, Y - Always, 1 - Once, or A-Auto)
 OPTHINT  . . . . . .          >
 PATH . . . . . . . .                                            >
 ENCODING . . . . . . 37       (ASCII, EBCDIC, UNICODE or ccsid)
 IMMEDWRITE . . . . . NO       (Yes,No or PH1)
 ROUNDING . . . . . . HALFEVEN  (Ceiling, Down, Floor, HalfDown,)
                                (HalfEven, HalfUp, or Up)
 CONCURRENTACCESSRES            (U - Usecurrentlycommitted or)
                                (W - Waitforoutcome)
)
```

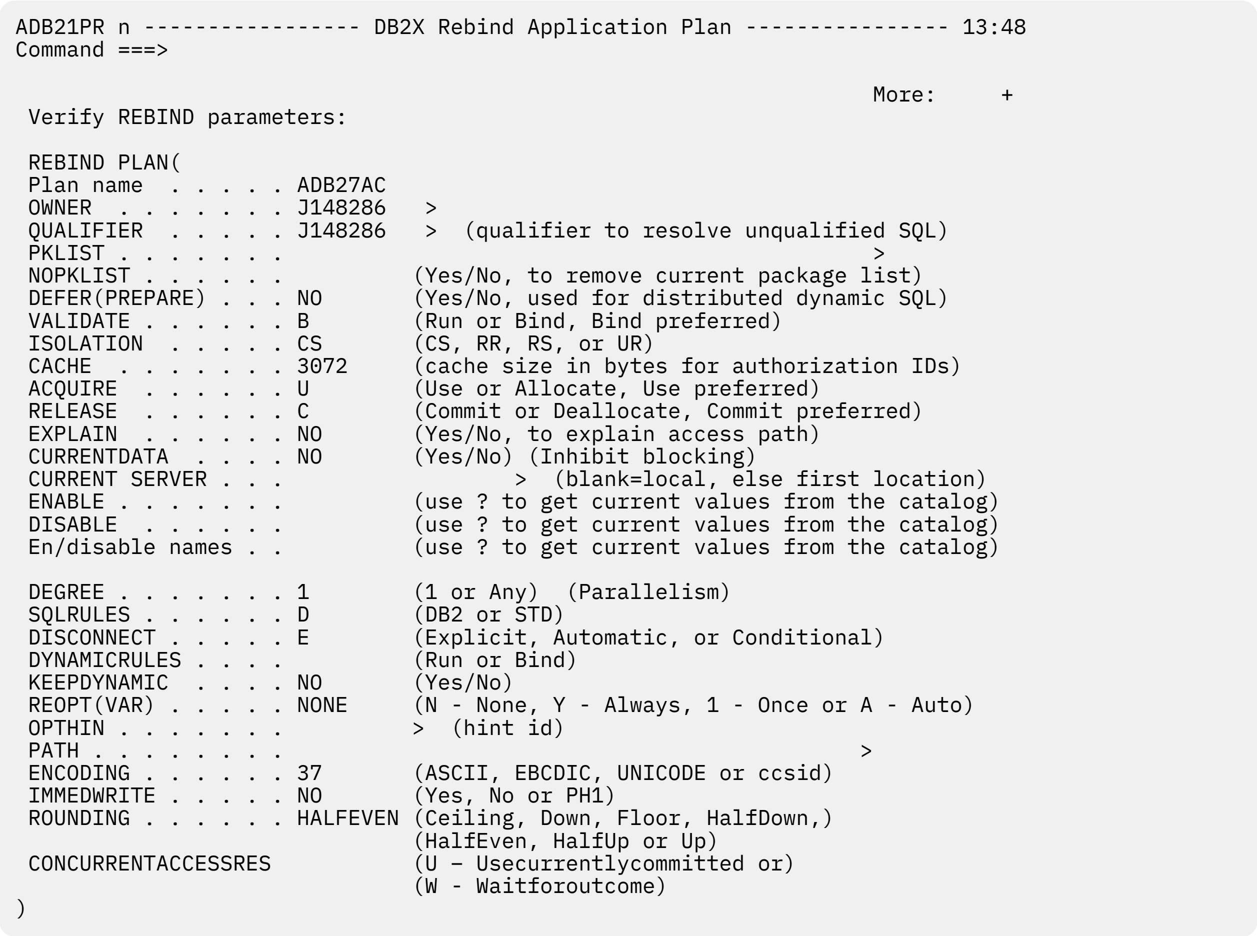
Figure 44. ***Bind Application Plan (ADB21PB)*** *panel*

Rebinding application plans

Rebind an application plan when changes have been made that affect the plan, but the SQL statements in the program have not changed.

To rebind an application plan:

1. On the **Application Plans (ADB21P)** panel, specify the RB line command (rebind plan) and press Enter.
2. On the **Rebind Application Plan (ADB21PR)** panel, enter your input and press Enter:



```
ADB21PR n ----------------- DB2X Rebind Application Plan ---------------- 13:48
Command ===>

                                                                  More:     +
 Verify REBIND parameters:

 REBIND PLAN(
 Plan name  . . . . . ADB27AC
 OWNER  . . . . . . . J148286   >
 QUALIFIER  . . . . . J148286   >  (qualifier to resolve unqualified SQL)
 PKLIST . . . . . . .                                              >
 NOPKLIST . . . . . .          (Yes/No, to remove current package list)
 DEFER(PREPARE) . . . NO       (Yes/No, used for distributed dynamic SQL)
 VALIDATE . . . . . . B        (Run or Bind, Bind preferred)
 ISOLATION  . . . . . CS       (CS, RR, RS, or UR)
 CACHE  . . . . . . . 3072     (cache size in bytes for authorization IDs)
 ACQUIRE  . . . . . . U        (Use or Allocate, Use preferred)
 RELEASE  . . . . . . C        (Commit or Deallocate, Commit preferred)
 EXPLAIN  . . . . . . NO       (Yes/No, to explain access path)
 CURRENTDATA  . . . . NO       (Yes/No) (Inhibit blocking)
 CURRENT SERVER . . .                  >  (blank=local, else first location)
 ENABLE . . . . . . .          (use ? to get current values from the catalog)
 DISABLE  . . . . . .          (use ? to get current values from the catalog)
 En/disable names . .          (use ? to get current values from the catalog)

 DEGREE . . . . . . . 1        (1 or Any)  (Parallelism)
 SQLRULES . . . . . . D        (DB2 or STD)
 DISCONNECT . . . . . E        (Explicit, Automatic, or Conditional)
 DYNAMICRULES . . . .          (Run or Bind)
 KEEPDYNAMIC  . . . . NO       (Yes/No)
 REOPT(VAR) . . . . . NONE     (N - None, Y - Always, 1 - Once or A - Auto)
 OPTHIN . . . . . . .          >  (hint id)
 PATH . . . . . . . .                                             >
 ENCODING . . . . . . 37       (ASCII, EBCDIC, UNICODE or ccsid)
 IMMEDWRITE . . . . . NO       (Yes, No or PH1)
 ROUNDING . . . . . . HALFEVEN (Ceiling, Down, Floor, HalfDown,)
                               (HalfEven, HalfUp or Up)
 CONCURRENTACCESSRES           (U – Usecurrentlycommitted or)
                               (W - Waitforoutcome)
)
```

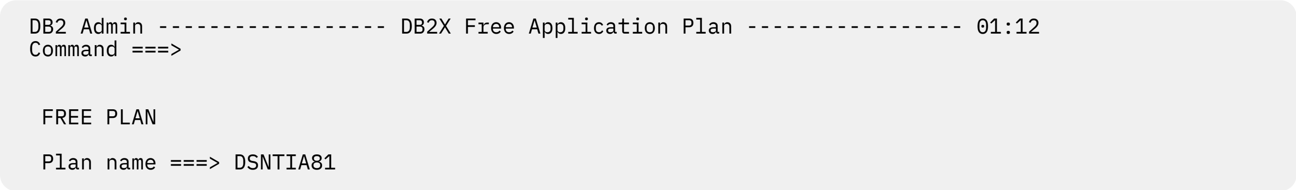
*Figure 45. **Rebind Application Plan (ADB21PR)** panel*

Freeing application plans

When you free an application plan, you delete it from Db2.

To free an application plan:

1. On the **Application Plans (ADB21P)** panel, specify the F line command (free plan) and press Enter.
2. On the **Free Application Plan (ADB21PF)** panel, enter your input and press Enter:



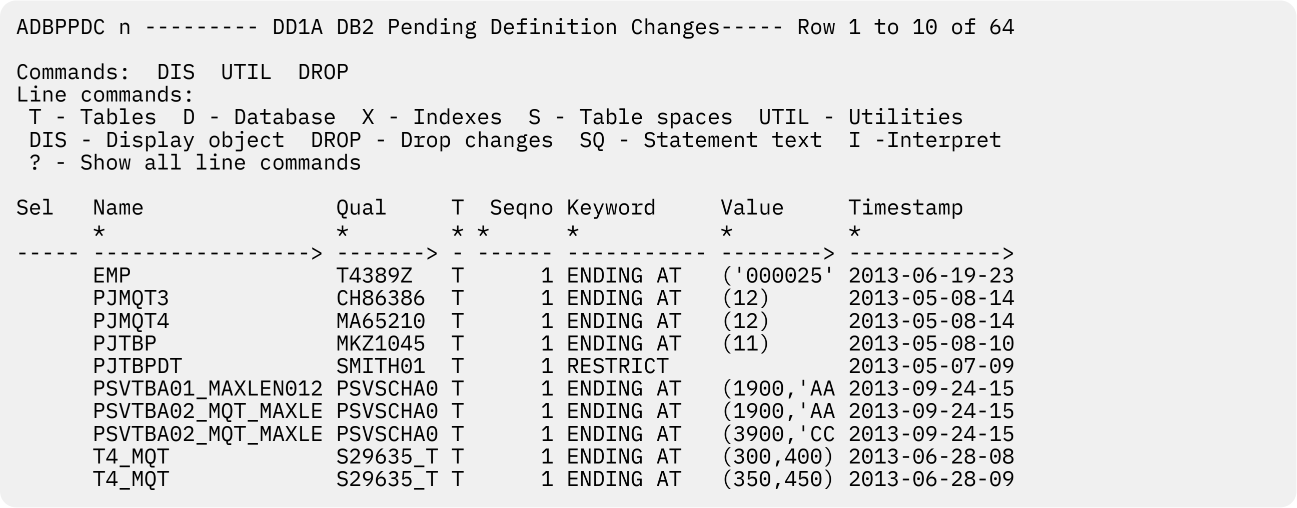
```
DB2 Admin ------------------ DB2X Free Application Plan ----------------- 01:12
Command ===>


 FREE PLAN

 Plan name ===> DSNTIA81
```

*Figure 46. **Free Application Plan (ADB21PF)** panel*

Option PDC. DB2 Pending Definition Changes

Use the **DB2 Pending Definition Changes** panel to display information about the definition changes that are pending in the Db2 catalog.

Select option PDC on the **System Catalog** panel to display the **DB2 Pending Definition Changes** panel, as shown in the following figure.



```
ADBPPDC n --------- DD1A DB2 Pending Definition Changes----- Row 1 to 10 of 64

Commands:  DIS  UTIL  DROP
Line commands:
 T - Tables  D - Database  X - Indexes  S - Table spaces  UTIL - Utilities
 DIS - Display object  DROP - Drop changes  SQ - Statement text  I -Interpret
 ? - Show all line commands

Sel   Name               Qual     T  Seqno Keyword     Value     Timestamp
      *                  *        * *      *           *         *
----- -----------------> -------> - ------ ----------- --------> ------------>
      EMP                T4389Z   T      1 ENDING AT   ('000025' 2013-06-19-23
      PJMQT3             CH86386  T      1 ENDING AT   (12)      2013-05-08-14
      PJMQT4             MA65210  T      1 ENDING AT   (12)      2013-05-08-14
      PJTBP              MKZ1045  T      1 ENDING AT   (11)      2013-05-08-10
      PJTBPDT            SMITH01  T      1 RESTRICT              2013-05-07-09
      PSVTBA01_MAXLEN012 PSVSCHA0 T      1 ENDING AT   (1900,'AA 2013-09-24-15
      PSVTBA02_MQT_MAXLE PSVSCHA0 T      1 ENDING AT   (1900,'AA 2013-09-24-15
      PSVTBA02_MQT_MAXLE PSVSCHA0 T      1 ENDING AT   (3900,'CC 2013-09-24-15
      T4_MQT             S29635_T T      1 ENDING AT   (300,400) 2013-06-28-08
      T4_MQT             S29635_T T      1 ENDING AT   (350,450) 2013-06-28-09
```

Figure 47. DB2 Pending Definition Changes panel (ADBPPDC)

The following primary commands are valid on this panel:

DIS
Performs Db2 DISPLAY command on the listed objects.

UTIL
Generates a utility JCL for all table spaces.

DROP
Drops the pending Db2 changes that are listed.

The following columns are on this panel:

Sel
Input field where you enter one of the line commands listed on the panel.

Name
Name of the object that has pending changes.

Qual
For a table space, the qualifier is the database name. For an index or table, the qualifier is the schema name.

T
Type of object, which is one of the following values:

S
Table space.

I
Index.

T
Table.

Seqno
The sequence number.

Keyword
The keyword of a pending change.

Value
This field shows the text of the value in the pending change.

If the text is truncated, type EXPAND on the primary command line, position the cursor on the default text field, and press Enter to display all of the text.

Timestamp
This field indicates when then the pending change was created.

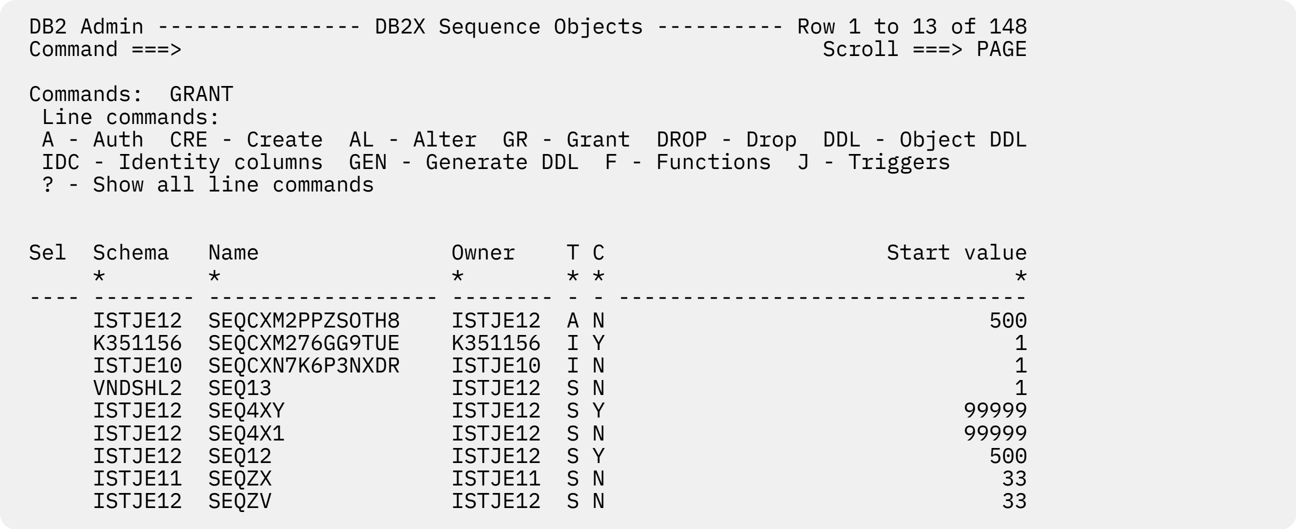
Option Q. Sequences

The **Sequence Objects** panel displays the sequences in the Db2 catalog.

A *sequence* is a user-defined object that generates a sequence of numeric values according to the specification with which the sequence was created. It efficiently provides recoverable, guaranteed-unique, sequential numbers to Db2 applications.

Select option Q on the **System Catalog** panel to display the **Sequence Objects** panel, as shown in the following figure.

On the **Sequence Objects** panel, you can issue the GEN primary command to generate SQL from Db2 catalog for all displayed sequences. You can also issue the GRANT primary command to change authorizations for all displayed sequences.



```
DB2 Admin ---------------- DB2X Sequence Objects ---------- Row 1 to 13 of 148
Command ===>                                                  Scroll ===> PAGE

Commands:  GRANT
 Line commands:
 A - Auth  CRE - Create  AL - Alter  GR - Grant  DROP - Drop  DDL - Object DDL
 IDC - Identity columns  GEN - Generate DDL  F - Functions  J - Triggers
 ? - Show all line commands


Sel  Schema   Name               Owner    T C                       Start value
     *        *                  *        * *                                 *
---- -------- ------------------ -------- - - -------------------------------
     ISTJE12  SEQCXM2PPZSOTH8    ISTJE12  A N                               500
     K351156  SEQCXM276GG9TUE    K351156  I Y                                 1
     ISTJE10  SEQCXN7K6P3NXDR    ISTJE10  I N                                 1
     VNDSHL2  SEQ13              ISTJE12  S N                                 1
     ISTJE12  SEQ4XY             ISTJE12  S Y                             99999
     ISTJE12  SEQ4X1             ISTJE12  S N                             99999
     ISTJE12  SEQ12              ISTJE12  S Y                               500
     ISTJE11  SEQZX              ISTJE11  S N                                33
     ISTJE12  SEQZV              ISTJE12  S N                                33
```

Figure 48. Sequence Objects panel (ADB21Q)

The following primary command is valid on this panel:

GRANT
Issues a GRANT command on multiple sequences.

Tip: The GRANT command operates on each row that is displayed in the table. If you want to omit some of the rows before you issue the command, you can use the minus (-) line command to remove rows from the display. The GRANT command operates only on rows that are listed.

The following columns are on this panel:

Sel
Input field in which you can enter a line command.

Schema
The schema of the sequence.

Name
Name of the sequence.

Owner
Owner of the sequence.

T
The sequence type. Possible values are:

S
User-defined sequence.

I
Identity column.

X
DOCID column for base table containing XML data.

A
Alias.

C
Specifies whether to wrap values after reaching the maximum value (maxvalue) or minimum value (minvalue).

Y
Yes.

N
No.

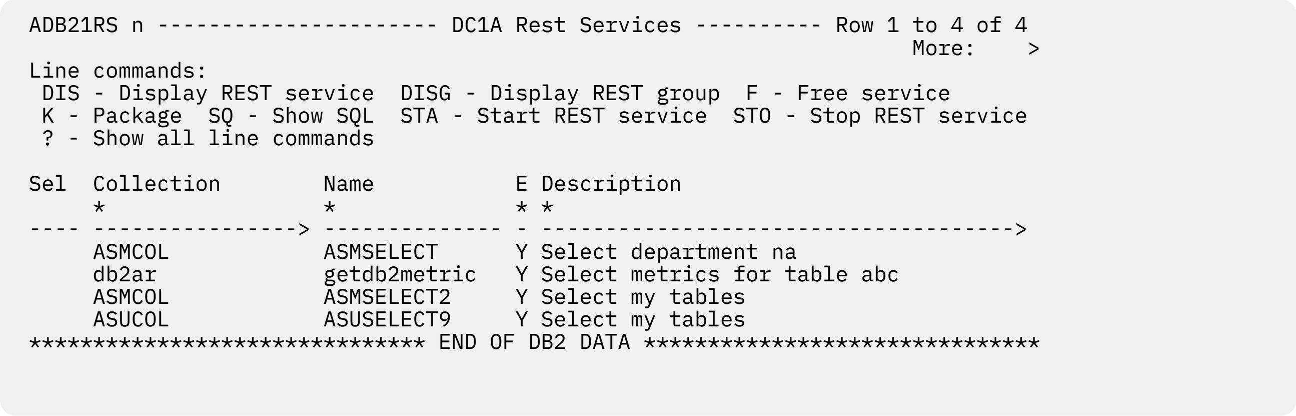
Start value
Indicates the first value for the sequence.

Option RS. REST Services

The **REST Services (ADB21RS)** panel displays information about Db2 REST services.

To display the **REST Services (ADB21RS)** panel, select option RS on the **System Catalog (ADB21)** panel. If no REST services exist, the **Create Rest Service (ADB21RSB)** panel is displayed instead so that you can first create a service.

Tip: If you receive SQLCODE -551 when you specify RS, ensure that you are authorized to access to SYSIBM.DSNSERVICE. See "Managing Db2 REST services" on page 872.



```
ADB21RS n ---------------------- DC1A Rest Services ---------- Row 1 to 4 of 4
                                                                     More:    >
Line commands:
 DIS - Display REST service  DISG - Display REST group  F - Free service
 K - Package  SQ - Show SQL  STA - Start REST service  STO - Stop REST service
 ? - Show all line commands

Sel  Collection        Name           E Description
     *                 *              * *
---- ----------------> ------------- - ------------------------------------->
     ASMCOL            ASMSELECT      Y Select department na
     db2ar             getdb2metric   Y Select metrics for table abc
     ASMCOL            ASMSELECT2     Y Select my tables
     ASUCOL            ASUSELECT9     Y Select my tables
******************************* END OF DB2 DATA *******************************
```

*Figure 49. **REST Services (ADB21RS)** panel*

Panel columns

This panel includes the following columns:

Sel
Select column. Use this input field to enter line commands.

Collection
The name of the collection that contains the package.

Name
The name of the package that contains the service request.

E
An indication of whether the service is enabled. Possible values are **Y** (yes) and **N** (no).

Description
A user-specified character string to describe the service.

Version
A user-specified character string that specifies the REST service version name.

This column is displayed only if Db2 REST services versioning support is enabled.

D
An indication of whether the REST service version that is represented by this row is the default version.

This column is displayed only if Db2 REST services versioning support is enabled.

Contoken
A consistency token for the package that is generated when the service is created or altered.

Createdts
The time when the row was inserted.

Alteredts
The time when the row was last updated.

Panel line commands

The following line commands are valid on this panel:

B
Bind the REST service.

CRE
Create a REST service.

DIS
Display REST service status.

DISG
Display REST service group status.

F
Free the package that contains the service request.

K
Show information about the package that contains the service request.

SQ
Show the SQL statement in the package.

STA
Start the REST service.

STAG
Start the REST service group.

STO
Stop the REST service.

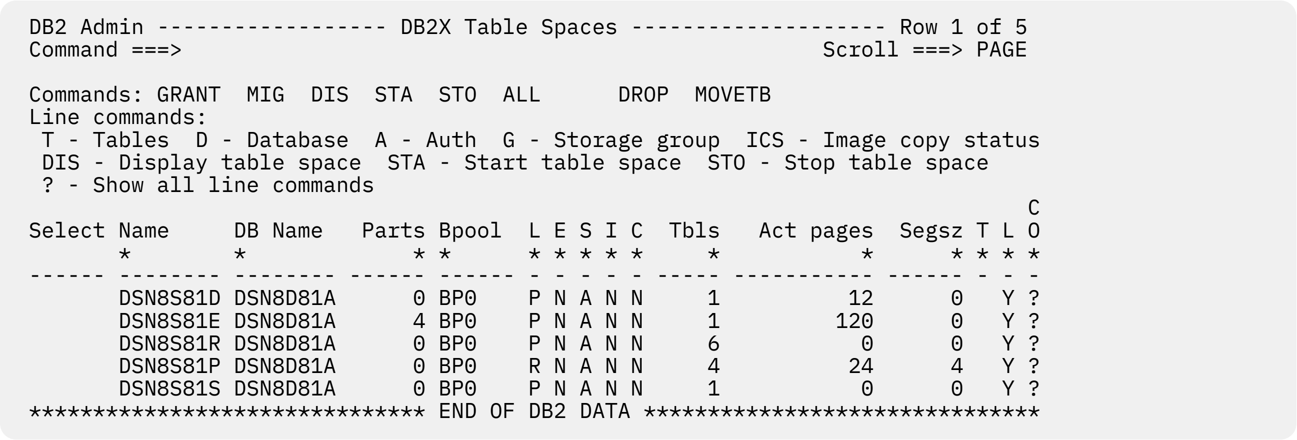
STOG
Stop the REST service group.

Related information
Db2 REST services (Db2 12 for z/OS)

Option S. Table Spaces

The **Table Spaces (ADB21S)** panel displays the table spaces in the Db2 catalog.

To display the **Table Spaces (ADB21S)** panel, select option S on the **System Catalog (ADB21)** panel.



```
DB2 Admin ------------------ DB2X Table Spaces -------------------- Row 1 of 5
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  ALL      DROP  MOVETB
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands
                                                                              C
Select Name     DB Name   Parts Bpool  L E S I C  Tbls   Act pages  Segsz T L O
       *        *             * *      * * * * *     *           *      * * * *
------ -------- -------- ------ ------ - - - - - ----- ----------- ------ - - -
       DSN8S81D DSN8D81A      0 BP0    P N A N N     1          12      0   Y ?
       DSN8S81E DSN8D81A      4 BP0    P N A N N     1         120      0   Y ?
       DSN8S81R DSN8D81A      0 BP0    P N A N N     6           0      0   Y ?
       DSN8S81P DSN8D81A      0 BP0    R N A N N     4          24      4   Y ?
       DSN8S81S DSN8D81A      0 BP0    P N A N N     1           0      0   Y ?
******************************* END OF DB2 DATA *******************************
```

Figure 50. ***Table Spaces (ADB21S)*** *panel*

The following primary commands are valid on this panel:

GRANT
Issues a GRANT command on multiple table spaces.

MIG
Issues a MIG command on multiple table spaces.

DIS
Issues a Db2 DISPLAY command on multiple table spaces.

STA
Issues a Db2 START command on multiple table spaces.

STO
Issues a Db2 STOP command on multiple table spaces.

ALL
Lists all objects of a specified type for each object in a list of objects. Supported values for this panel are:

ALL T
Shows all tables for the listed table spaces. Views or aliases are not shown.

ALL K
Shows all packages for the listed table spaces.

ALL X
Shows all indexes for the listed table spaces.

DROP
Drops all listed table spaces.

MOVETB
Moves tables from multi-table table spaces to partition-by-growth (PBG) universal table spaces (UTS).

If either of the following conditions are true, you are prompted to send the statements to a batch job or work statement list (WSL):

- The size of the statements generated by the GRANT, DIS, STA, or STO primary command exceeds 32K (an ISPF limit).
- The number of statements generated by the DIS, STA, or STO primary command exceeds 10

Restriction: The DROP line command does not allow implicit LOB table spaces to be dropped, but it does allow explicit LOB table spaces to be dropped. This restriction protects you from leaving a definition incomplete.

Recommendation: Primary commands operate on each row that is displayed in the table. If you want to omit some of the rows before you issue the primary command, use the minus (-) line command to remove rows from the display. The primary commands operate only on rows that are listed.

The panel displays the following columns:

Select
Input field where you enter one of the line commands listed on the panel.

Name
Name of the table space.

DB Name
Name of the database.

Parts
Number of partitions for a table space. For non-partitioned table spaces, this value is 0.

To display detailed information for a table space, issue the SP line command against that table space. To display the data set name for the table space (or the data set names for every partition of a partitioned table space), issue the DSN line command against that table space. You can also use the DSN line command against a single partition after you issue the SP line command to display the data set name for that partition only.

Bpool
Name of the buffer pool used for the table space.

L
Locking size. This column can contain one of the following values:

A
Any

L
Large object (LOB)

P
Page

R
Row

S
Table space

T
Table

X
Implicitly created XML table space

E
Erase rule. This column can contain one of the following values:

Y
Erase

N
No erase

S
Status of the table space. This column can contain one of the following values:

A
Available

C
Incomplete, part index

P
Check pending

S
Alt check pending

T
Incomplete, table

I
Implicit (whether the table space was created implicitly). This column can contain one of the following values:

Y
Yes

N
No

C
Close rule. This column can contain one of the following values:

Y
Yes

N
No

Tbls
Number of tables defined in the table space

If you are running a version of Db2 that is lower than Db2 12 function level 509, this column is called **Tables** instead.

Act. pages
Number of active pages in the table space. This field is 0 if the RUNSTATS utility has not been run.

Segsz
Number of pages in each segment of a segmented table space. The value is 0 if the table space is not segmented.

T
Type of table space. This column can contain one of the following values:

Blank
Normal

G
Defined with the MAXPARTITIONS option (a partitioned-by-growth table space) with the underlying structure of a universal table space (UTS)

I
Defined with MEMBER CLUSTER and is not greater than 64 GB

K
Defined with MEMBER CLUSTER and can be greater than 64 GB

L
Defined as LARGE and can be greater than 64 GB

O
Defined as an LOB (large object) table space

P
Implicit table space created for XML columns

R
Range-partitioned UTS

L
Log changes. This column can contain one of the following values:

Y
Yes

N
No

X
: This LOB or XML table space has the NOT LOGGED attribute. Undo and redo logging for the table space is suppressed. Also, the logging attribute for this LOB or XML table space is linked to the logging attribute of the associated base table space and might not be able to be altered independently. If the logging attribute of the base table space is altered to LOGGED, the logging attribute of the LOB or XML table space will also be altered to LOGGED.

CO

The data compression method that is used by the table space. This column is displayed only for Db2 12 function 509 and higher. It can contain one of the following values:

Y

The Db2 default compression algorithm

H

Huffman compression

F

Fixed-length compression

blank

Compression is not used.

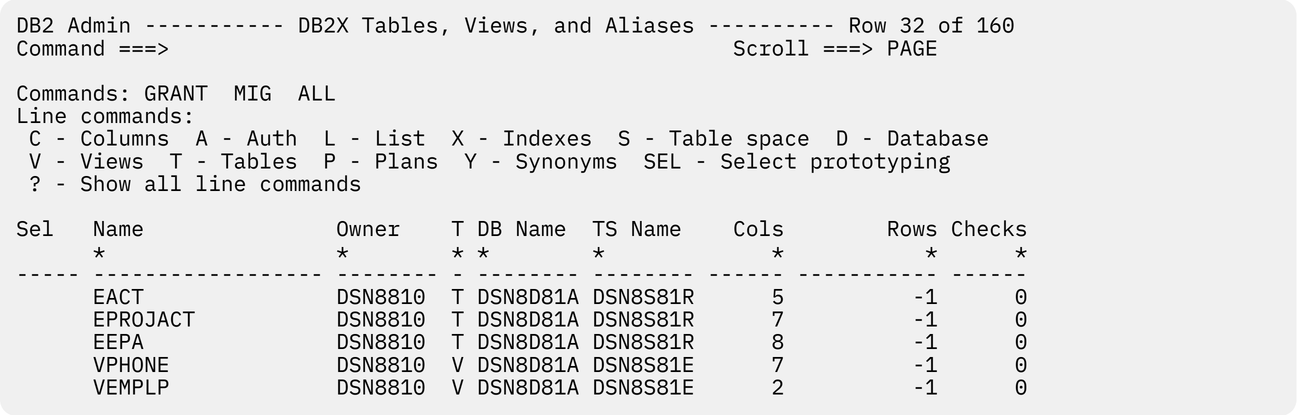
?

Null. This column contains the null value when the value is unknown for objects created prior to Db2 12.

Option T. Tables, Views, and Aliases

The **Tables, Views, and Aliases (ADB21T)** panel displays the tables, views, and aliases in the Db2 catalog.

Select option T on the **System Catalog (ADB21)** panel to display the **Tables, Views, and Aliases (ADB21T)** panel, as shown in the following figure.



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

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

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

*Figure 51. **Tables, Views, and Aliases (ADB21T)** panel*

On the **Tables, Views, and Aliases (ADB21T)** panel, you can issue many line commands. Enter a question mark (?) on a row to view all valid line commands. These line commands include:

N

Lists constraints on the table.

GEN

Enables you to reverse engineer Db2 objects from this panel.

MIG

Migrates tables and lists of tables.

UTL

Generates JCL that can be run against a table.

J
Displays triggers on the table. This line command works on views as well as tables.

XML
Displays the XML tables if the table has XML columns. For more information, see "Viewing XML tables" on page 181.

CLONE
Displays the clone table if the table is a base table with a defined clone. For more information, see "Viewing clone tables" on page 183.

The following primary commands are valid on this panel:

ALL
Lists all objects of a specified type for each object in a list of objects. Supported values for this panel are:

ALL S
Shows all table spaces for the listed tables.

ALL K
Shows all packages for the listed tables.

ALL X
Shows all indexes for the listed tables.

ALL A
Shows all aliases for the listed tables.

ALL V
Shows all first-level views for the listed tables. Views on views will not be shown.

ALL VV
Show all views for the listed tables, including views on views.

BET and DET
Generates a detail report for tables and related objects. DET generates the report online. BET generates the report in batch.

This primary command is available for the following table types:

- C: Clone table
- G: Created global temporary table
- H: History table
- P: Implicit table created for XML columns
- T: Table
- X: Auxiliary table

CT
Migrates objects by using Db2 Cloning Tool.

GEN
Generates SQL for the listed objects.

GRANT
Issues a GRANT command on multiple tables and views.

MIG
Issues a MIG command on multiple tables.

REP
Generates a report for the listed objects

UT or UTL or UTIL
Generates utility JCL for the listed objects.

Tip: Primary commands operate on each row that is displayed in the table. If you want to omit some of the rows before you issue the primary command, use the minus (-) line command to remove rows from the display. The primary commands operate only on rows that are listed.

This panel includes the following columns:

Sel
Input field where you enter one of the line commands listed on the panel

Name
Name of the table, view, or alias

Owner
Authorization ID of the owner of the table, view, or alias

T
Type of object, which is one of the following values:

T
Table

V
View

A
Alias

G
Global temporary table

X
Auxiliary table. If the base table containing LOB column(s) is dropped and recreated, the explicit auxiliary table is recreated according to its source definition. Changes to the auxiliary table are not reported. Updates to the auxiliary table are ignored if the base table is not recreated.

M
Materialized table. A materialized table is similar to a view, in that a full SELECT statement is used to create the materialized table query on a table or a view. A materialized table contains physical data behind it and is maintained by the system or by a user. You can use the REFRESH command to refresh the materialized table data. Only a user-maintained materialized table can contain inserts, deletes, and updates. **Restriction:** When a table contains materialized queries, no ALTER commands can be performed on that table.

You can create a materialized table using the CREM command against a table or a view. You can also create a new materialized table by issuing the CRE command against an existing materialized table.

You can alter a regular table to make it be a materialized table. Issue the ALM command against a regular table to change it to a materialized table. You can use the DROPM command against a materialized table to drop a materialized query from the materialized table, changing it to a regular table.

P
Implicit tables created for XML columns

C
Clone table

DB Name
For a table or a view of tables, the name of the database that contains the table space named in TS NAME field. For a view of a view, a global temporary table or for an alias, this field contains DSNDB06.

TS Name
For a table or a view of one table, the name of the table space that contains the table. For a view of a view, this field contains SYSVIEWS. For an alias, this field contains SYSDBAUT.

Cols
Number of columns in the table or view

Rows
Total number of rows in the table. If the RUNSTATS utility has not been run or if the rows describe a view or an alias, this field contains a value of -1.

Checks
Number of check constraints defined on the table

C
Access control enforced by: R - Row C - Col B - Both ' ' - NA

For information about other columns, see Db2 catalog tables (Db2 12 for z/OS).

Issuing table space commands

You can issue table space commands on the **Tables, Views, and Aliases (ADB21T)** panel by appending up to three characters of the table space command to the S command. For example, to display a table space, issue the S.DIS command against the table whose table space you want to display. In addition, the following table space commands can be issued from the **Tables, Views, and Aliases (ADB21T)** panel without prefacing with "S":

- ICS
- IDS
- DISA
- DISC
- DISL
- DISR
- DIST
- DISU
- STA
- STAFO
- STARO
- STARW
- STAUT
- U.C
- U.CC
- U.CI
- U.C2
- U.DG
- U.E
- U.EN
- U.K
- U.KD
- U.KL
- U.M
- U.N
- U.NA
- U.NB
- U.NC
- U.NL
- U.NR

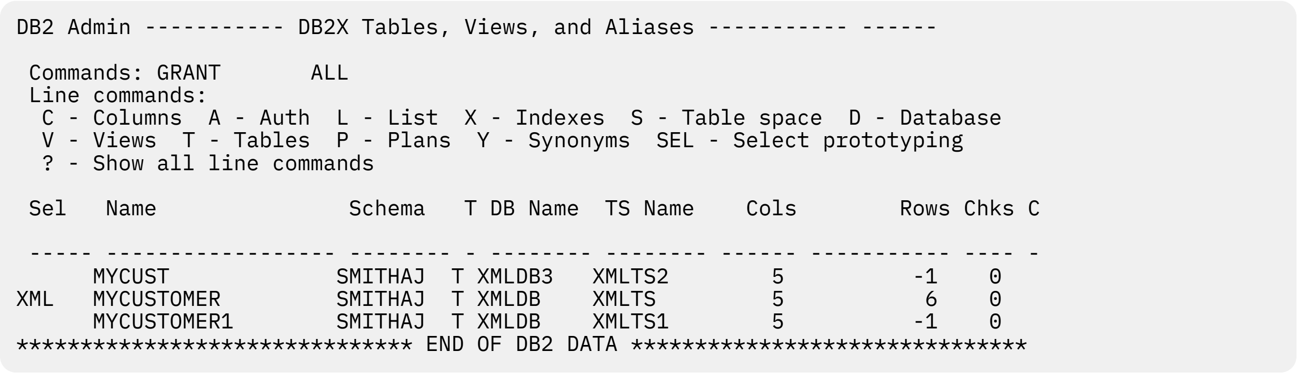
- U.NW
- U.NX
- U.O
- U.OC
- U.OO
- U.OU
- U.P
- U.Q
- U.R
- U.RR
- U.RT
- U.RX
- U.SM
- U.U
- U.V
- U.VC
- U.VG
- U.VI
- U.VL
- U.VP
- U.VR

To execute commands that are longer than three characters, such as STAFO, you must first issue the S.? command on the **Tables, Views, and Aliases (ADB21T)** panel. This command displays a list of all the valid table space commands. From this list, you can select commands to execute.

Note: Table space commands are not allowed on views.

Viewing XML tables

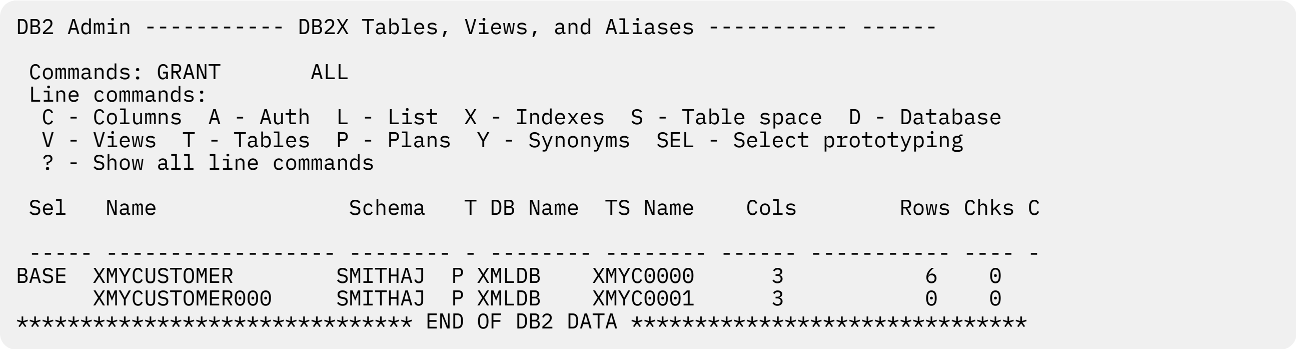
Use the XML line command against a table that has XML columns to display the XML tables. You issue the XML line command on the Tables, Views, and Aliases panel.



```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

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

 Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C

 ----- ------------------ -------- - -------- -------- ------ ----------- ---- -
      MYCUST             SMITHAJ  T XMLDB3   XMLTS2        5          -1    0
XML   MYCUSTOMER         SMITHAJ  T XMLDB    XMLTS         5           6    0
      MYCUSTOMER1        SMITHAJ  T XMLDB    XMLTS1        5          -1    0
******************************* END OF DB2 DATA *******************************
```

Figure 52. The Tables, Views, and Aliases panel (ADB21T) – viewing XML tables

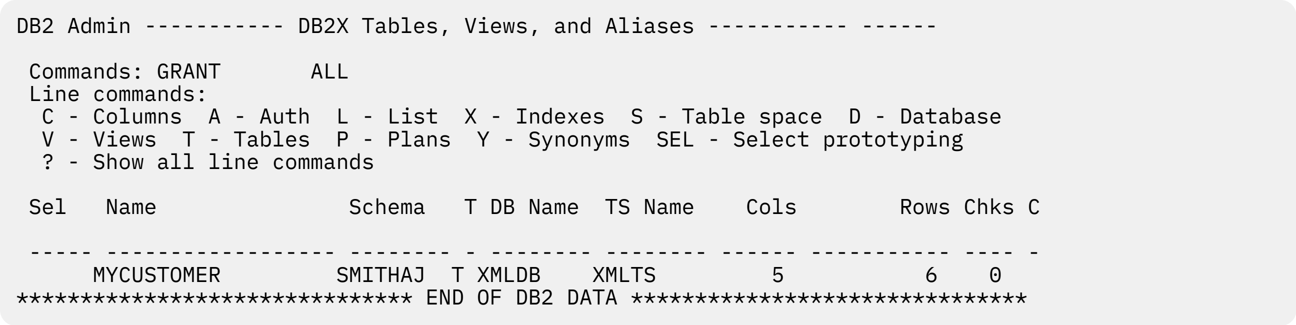
You can issue the BASE line command against an XML table to show its corresponding base table, as shown in the following figure:



```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

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

 Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C

 ----- ------------------ -------- - -------- -------- ------ ----------- ---- -
BASE  XMYCUSTOMER        SMITHAJ  P XMLDB    XMYC0000      3           6    0
      XMYCUSTOMER000     SMITHAJ  P XMLDB    XMYC0001      3           0    0
******************************* END OF DB2 DATA *******************************
```

Figure 53. The Tables, Views, and Aliases panel (ADB21T) – viewing XML base

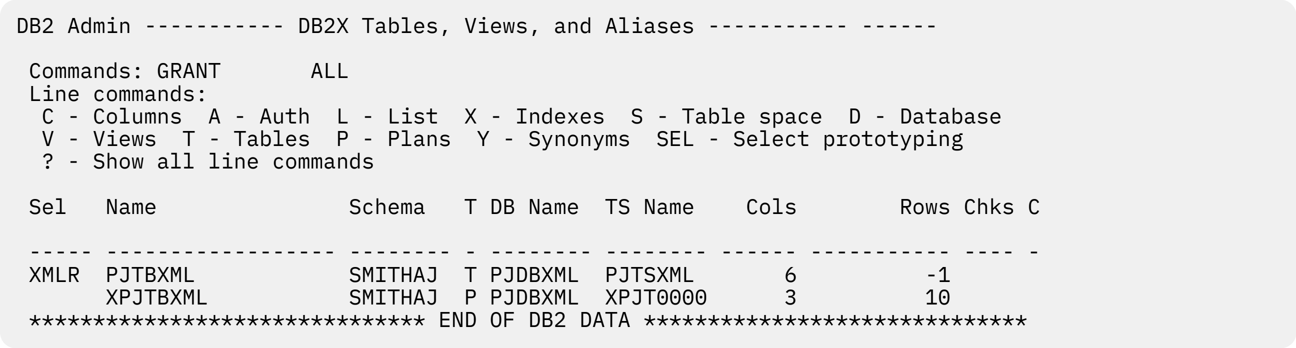
The corresponding base table is shown in the following figure:



```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

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

 Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C

 ----- ------------------ -------- - -------- -------- ------ ----------- ---- -
      MYCUSTOMER         SMITHAJ  T XMLDB    XMLTS         5           6    0
******************************* END OF DB2 DATA *******************************
```

Figure 54. The Tables, Views, and Aliases panel (ADB21T) – viewing XML base 2

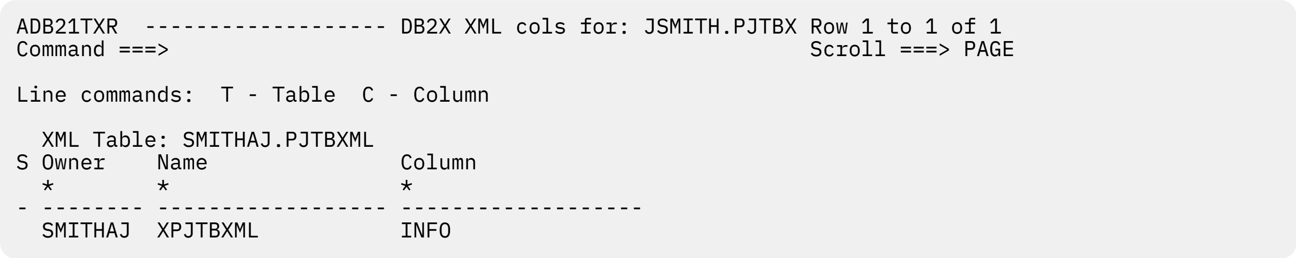
Issue the XMLR line command against a base table that has XML columns to display information about the XML columns and the related XML base table.



```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

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

 Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C

 ----- ------------------ -------- - -------- -------- ------ ----------- ---- -
 XMLR  PJTBXML            SMITHAJ  T PJDBXML  PJTSXML       6          -1
       XPJTBXML           SMITHAJ  P PJDBXML  XPJT0000      3          10
 ******************************* END OF DB2 DATA ******************************
```

Figure 55. The Tables, Views, and Aliases panel (ADB21T) – viewing XML column information

The following panel shows the XML column information and the related XML base table.

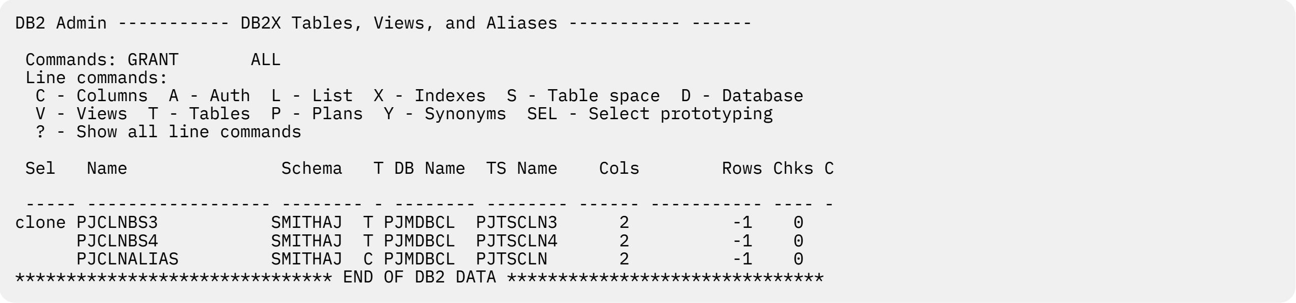


```
ADB21TXR  ------------------- DB2X XML cols for: JSMITH.PJTBX Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Line commands:  T - Table  C - Column

  XML Table: SMITHAJ.PJTBXML
S Owner    Name               Column
  *        *                  *
- -------- ----------------- ------------------
  SMITHAJ  XPJTBXML           INFO
```

Figure 56. The XML cols panel (ADB21TXR) – XML table column information 2

Viewing clone tables

Use the CLONE line command against a table that has a defined clone to display the clone table. You issue the CLONE line command on the Tables, Views, and Aliases panel.



```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

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

 Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C

 ----- ------------------ -------- - -------- -------- ------ ----------- ---- -
clone PJCLNBS3           SMITHAJ  T PJMDBCL  PJTSCLN3      2          -1    0
      PJCLNBS4           SMITHAJ  T PJMDBCL  PJTSCLN4      2          -1    0
      PJCLNALIAS         SMITHAJ  C PJMDBCL  PJTSCLN       2          -1    0
******************************* END OF DB2 DATA *******************************
```

Figure 57. The Tables, Views, and Aliases panel (ADB21T) – viewing clone tables

The following additional line commands support clone tables:

BASE
Shows the base table for a clone.

DROP
Drop clone tables.

XCHG
Exchange data between base and clone tables.

Note: To see the complete set of line commands for clone tables, enter the "? - Show all line commands" line command on the ADB21T panel.

Option TR. Trusted Contexts

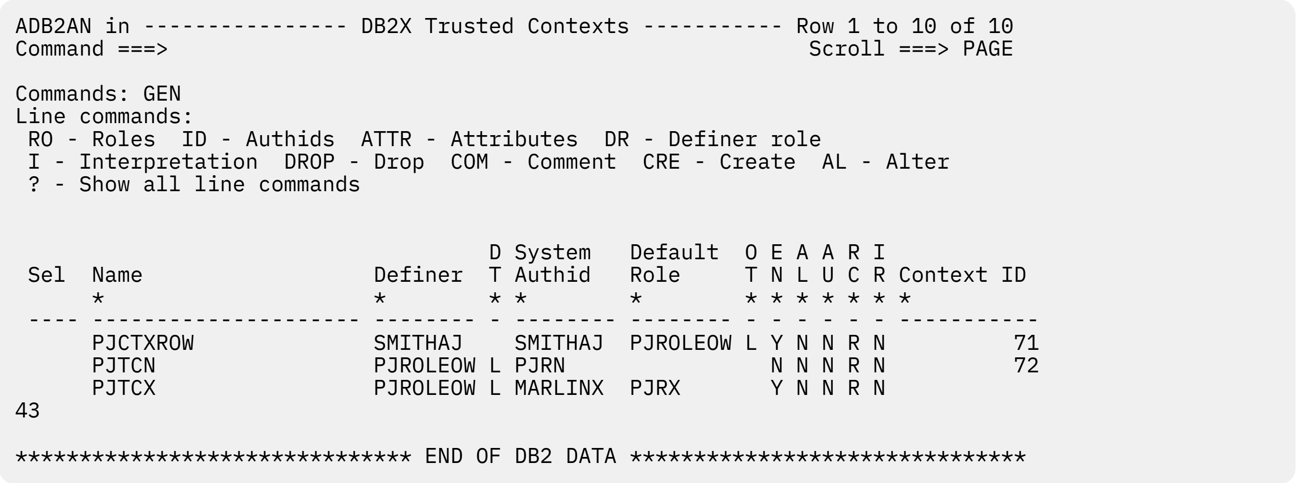
To display trusted contexts choose the TR option on the **System Catalog** panel.

The trusted contexts panel

Select option 1 on the DB2 Administration Menu to display the **System Catalog** panel. Select option AO, and then from the **Authorization Options** panel, select option TR to access the panel for trusted contexts.

The trusted contexts are shown in the following figure.

Note: The only selection criteria allowed for RO and TR options is Name and Column/Operator/Value.



```
ADB2AN in ---------------- DB2X Trusted Contexts ----------- Row 1 to 10 of 10
Command ===>                                                Scroll ===> PAGE

Commands: GEN
Line commands:
 RO - Roles  ID - Authids  ATTR - Attributes  DR - Definer role
 I - Interpretation  DROP - Drop  COM - Comment  CRE - Create  AL - Alter
 ? - Show all line commands


                                   D System   Default  O E A A R I
 Sel  Name                 Definer  T Authid   Role     T N L U C R Context ID
      *                    *        * *        *        * * * * * *
 ---- -------------------- -------- - -------- -------- - - - - - - -----------
      PJCTXROW             SMITHAJ    SMITHAJ  PJROLEOW L Y N N R N          71
      PJTCN                PJROLEOW L PJRN                N N N R N          72
      PJTCX                PJROLEOW L MARLINX  PJRX       Y N N R N
43

******************************* END OF DB2 DATA *******************************
```

Figure 58. Trusted Contexts panel (ADB2AN)

Use the following line commands from this panel to display trusted contexts information:

RO
Displays the default role, if any, and any roles from associated authorization IDs (panel ADB2ARL)

ID
Displays authorization IDs associated with a trusted context (panel ADB2ANID)

ATTR
Displays trusted context attributes (panel ADB2ANAT)

DR
Displays the role which defined the trusted context,if any (panel ADB2ARL)

I
Displays interpretation of an object in SYSCONTEXT (panel ADB2ANI)

DROP
Use to DROP a trusted context or attribute (panel ADB26DR)

COM
Allows you to create a comment for the trusted context (panel ADB26RT)

CRE
Use to create a trusted context (panel ADB26CN)

AL
Use to alter a trusted context (panel ADB26CN)

ADDA
Use to add an attribute to a trusted context (panel ADB26CN)

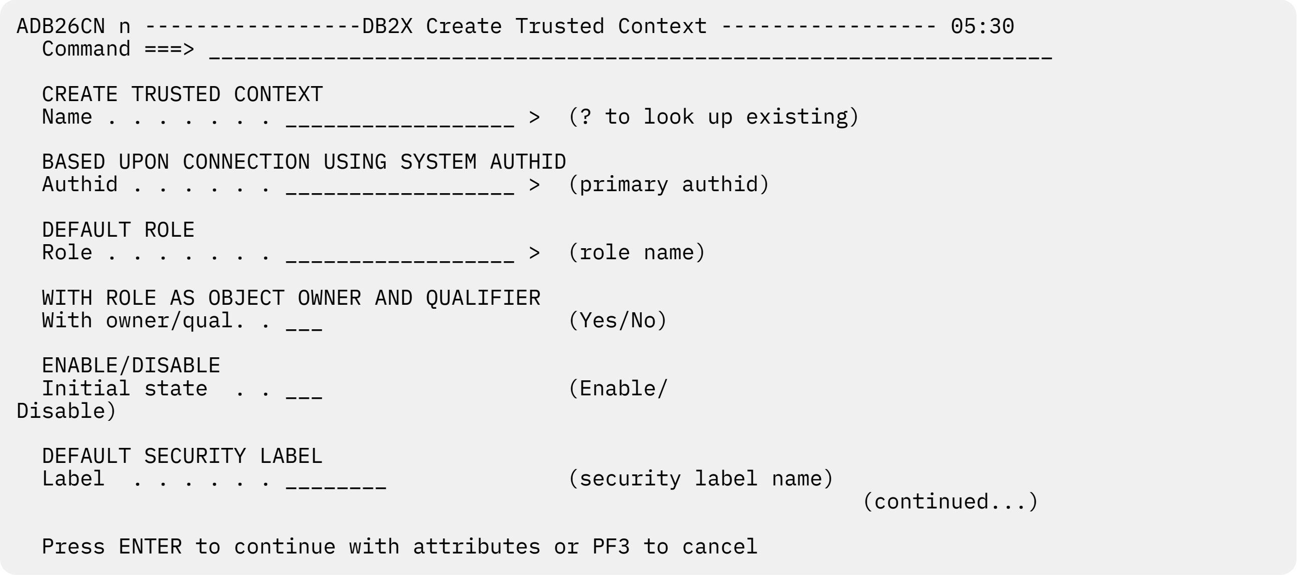
ADDI
Use to add an AuthID to a trusted context (panel ADB26CN)

DDL
Use to generate DDL

GEN
Use to generate SQL from Db2 catalog

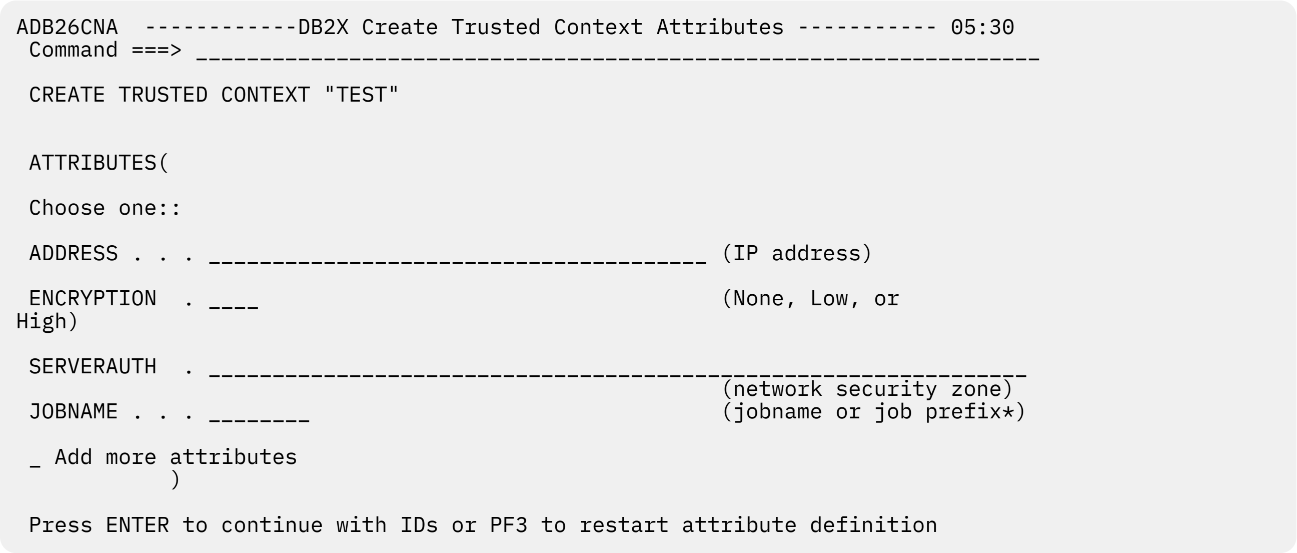
Creating or altering a trusted context

To create a trusted context, enter the CRE line command on panel ADB2AN. To alter a trusted context, enter the AL line command on panel ADB2AN. Fill in the required information in the series of panels that appear (shown in the following figure). An example is given for the CRE command.



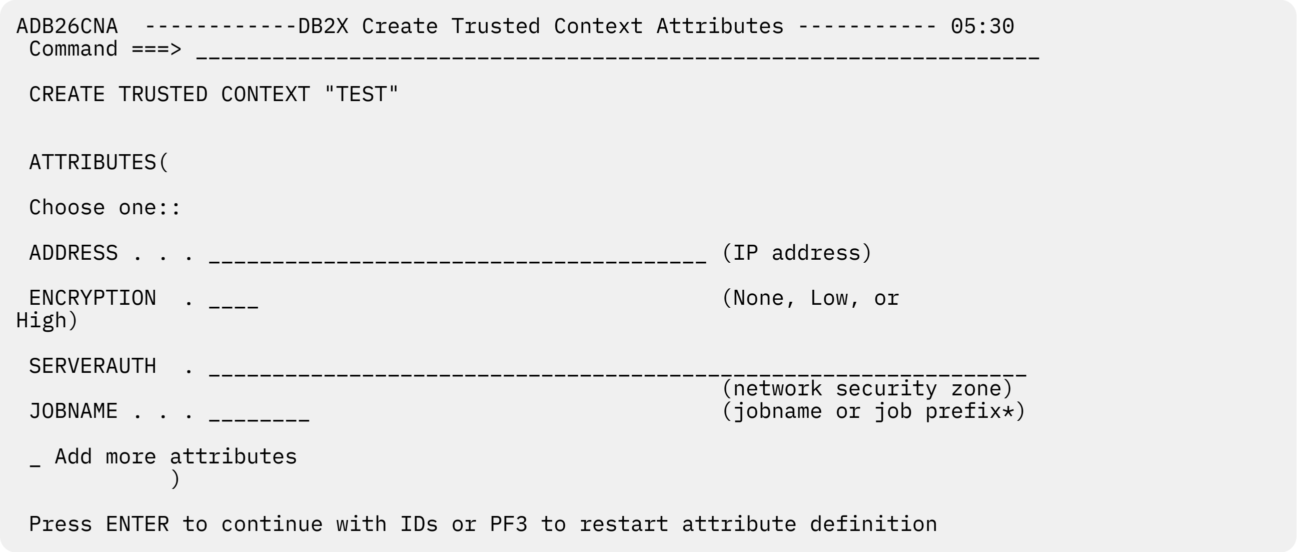
```
ADB26CN n -----------------DB2X Create Trusted Context ----------------- 05:30
  Command ===> __________________________________________________________________

  CREATE TRUSTED CONTEXT
  Name . . . . . . . __________________ >  (? to look up existing)

  BASED UPON CONNECTION USING SYSTEM AUTHID
  Authid . . . . . . __________________ >  (primary authid)

  DEFAULT ROLE
  Role . . . . . . . __________________ >  (role name)

  WITH ROLE AS OBJECT OWNER AND QUALIFIER
  With owner/qual. . ___                   (Yes/No)

  ENABLE/DISABLE
  Initial state  . . ___                   (Enable/
Disable)

  DEFAULT SECURITY LABEL
  Label  . . . . . . ________              (security label name)
                                                                  (continued...)

  Press ENTER to continue with attributes or PF3 to cancel
```

Figure 59. Create Trusted Contexts panel (ADB26CN)



```
ADB26CNA  ------------DB2X Create Trusted Context Attributes ----------- 05:30
 Command ===> __________________________________________________________________

 CREATE TRUSTED CONTEXT "TEST"


 ATTRIBUTES(

 Choose one::

 ADDRESS . . . _______________________________________ (IP address)

 ENCRYPTION  . ____                                    (None, Low, or
High)

 SERVERAUTH  . ________________________________________________________________
                                                       (network security zone)
 JOBNAME . . . ________                                (jobname or job prefix*)

 _ Add more attributes
            )

 Press ENTER to continue with IDs or PF3 to restart attribute definition
```

Figure 60. Create Trusted Context Attribues (ADB26CNA)



```
ADB26CNA  ------------DB2X Create Trusted Context Attributes ----------- 05:30
 Command ===> __________________________________________________________________

 CREATE TRUSTED CONTEXT "TEST"


 ATTRIBUTES(

 Choose one::

 ADDRESS . . . _______________________________________ (IP address)

 ENCRYPTION  . ____                                    (None, Low, or
High)

 SERVERAUTH  . ________________________________________________________________
                                                       (network security zone)
 JOBNAME . . . ________                                (jobname or job prefix*)

 _ Add more attributes
            )

 Press ENTER to continue with IDs or PF3 to restart attribute definition
```

Figure 61. Create Trusted Context IDs (ADB26CNU)

Option V. Views

The **Views (ADB21VV)** panel displays information about views, including whether the view uses multiple tables in different databases or table spaces. Alternatively, you can see information about views on the **Tables, Views, and Aliases (ADB21T)** panel. However, that panel does not show whether multiple databases or table spaces are used by the view.

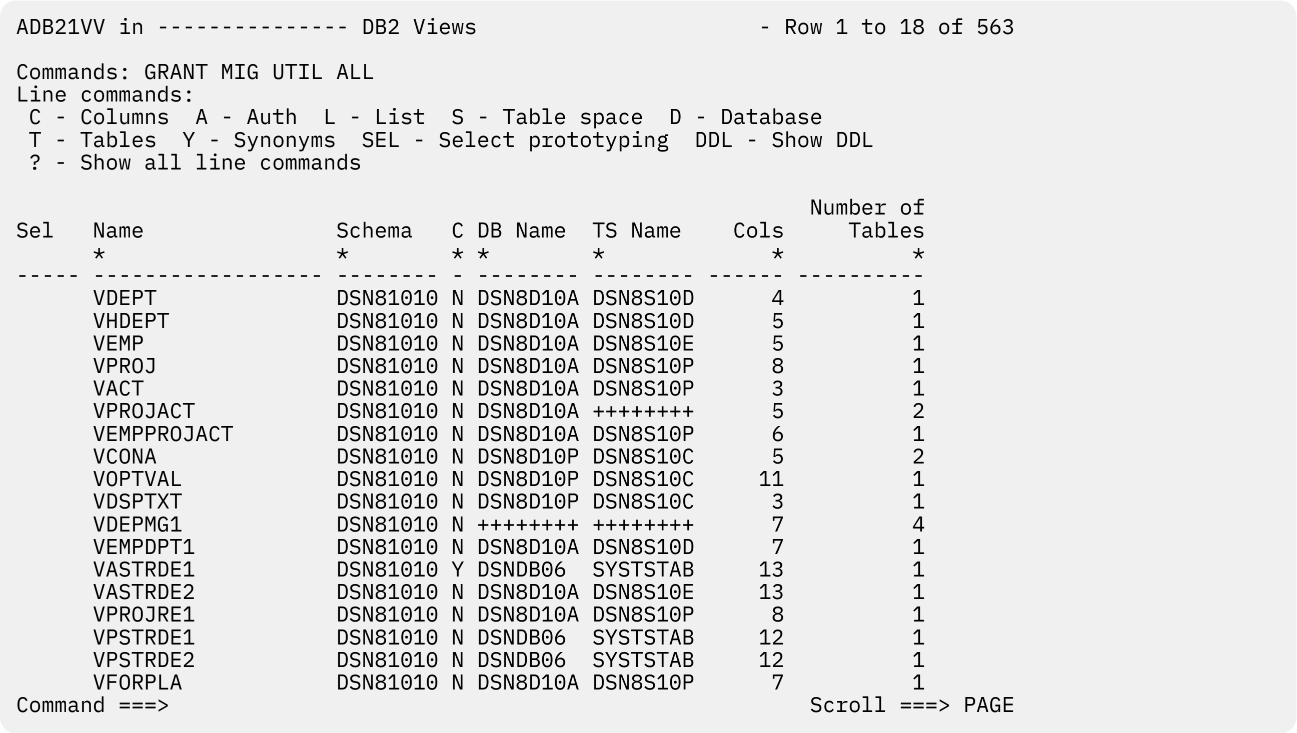
Displaying views by using option V - Views (ADB21VV) panel

On the **System Catalog (ADB21)** panel, select option V to display the **Views (ADB21VV)** panel.

On this panel, the TBNAME and DBNAME fields contain the following characters if the view has multiple tables in more than one table space or database or if the view references another view or a materialized query table (MQT):

```
++++++++
```

Use line commands to show all Db2 objects on which the view depends and any dependent views (a view of a view).



```
ADB21VV in --------------- DB2 Views                      - Row 1 to 18 of 563

Commands: GRANT MIG UTIL ALL
Line commands:
 C - Columns  A - Auth  L - List  S - Table space  D - Database
 T - Tables  Y - Synonyms  SEL - Select prototyping  DDL - Show DDL
 ? - Show all line commands

                                                            Number of
Sel   Name               Schema   C DB Name  TS Name    Cols    Tables
      *                  *        * *        *             *          *
----- ------------------ -------- - -------- -------- ------ ----------
      VDEPT              DSN81010 N DSN8D10A DSN8S10D      4          1
      VHDEPT             DSN81010 N DSN8D10A DSN8S10D      5          1
      VEMP               DSN81010 N DSN8D10A DSN8S10E      5          1
      VPROJ              DSN81010 N DSN8D10A DSN8S10P      8          1
      VACT               DSN81010 N DSN8D10A DSN8S10P      3          1
      VPROJACT           DSN81010 N DSN8D10A ++++++++      5          2
      VEMPPROJACT        DSN81010 N DSN8D10A DSN8S10P      6          1
      VCONA              DSN81010 N DSN8D10P DSN8S10C      5          2
      VOPTVAL            DSN81010 N DSN8D10P DSN8S10C     11          1
      VDSPTXT            DSN81010 N DSN8D10P DSN8S10C      3          1
      VDEPMG1            DSN81010 N ++++++++ ++++++++      7          4
      VEMPDPT1           DSN81010 N DSN8D10A DSN8S10D      7          1
      VASTRDE1           DSN81010 Y DSNDB06  SYSTSTAB     13          1
      VASTRDE2           DSN81010 N DSN8D10A DSN8S10E     13          1
      VPROJRE1           DSN81010 N DSN8D10A DSN8S10P      8          1
      VPSTRDE1           DSN81010 N DSNDB06  SYSTSTAB     12          1
      VPSTRDE2           DSN81010 N DSNDB06  SYSTSTAB     12          1
      VFORPLA            DSN81010 N DSN8D10A DSN8S10P      7          1
Command ===>                                                Scroll ===> PAGE
```

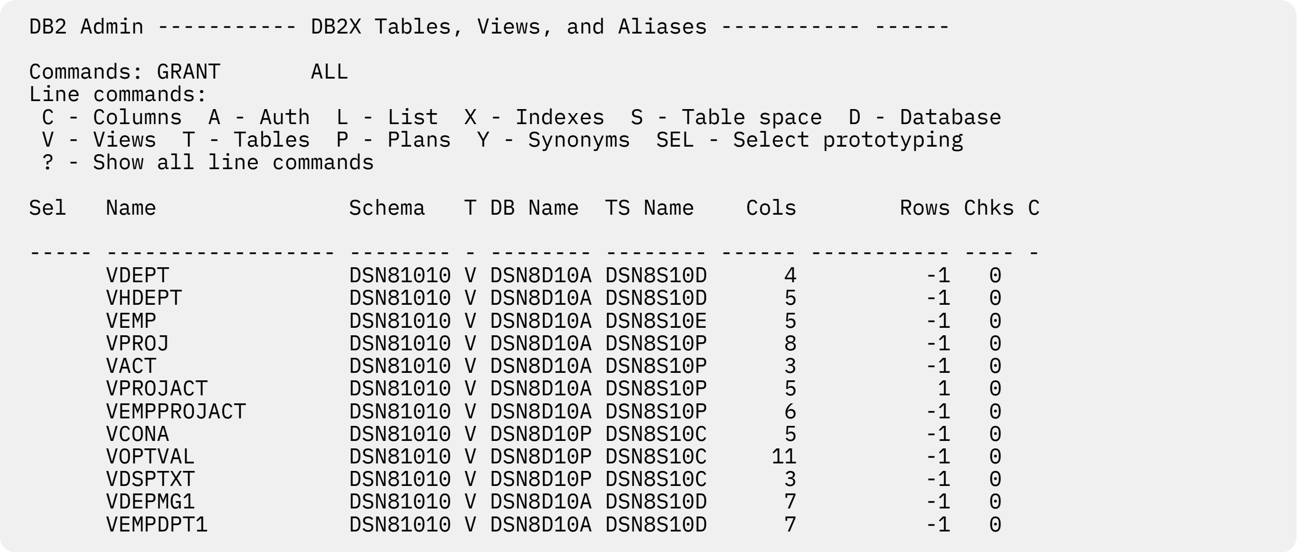
*Figure 62. **Views (ADB21VV)** panel*

Displaying views by using option TV - Tables, Views, and Aliases (ADB21T) panel

On the **System Catalog (ADB21)** panel, select option TV to display the **Tables, Views, and Aliases (ADB21T)** panel with a filter showing only views in the catalog.

This panel displays data from the SYSTABLES catalog table. In the case of a view that uses multiple tables from different databases or different table spaces (for example, in a join), the DBNAME and TSNAME fields in SYSTABLES include only one of the database or table space names from one of those tables. Therefore, on the **Tables, Views, and Aliases (ADB21T)** panel, you cannot tell whether the DBNAME or TSNAME values apply to all of the tables that are used by the view or for only one table in a join.

Scroll right (PF11) to view the following additional columns.



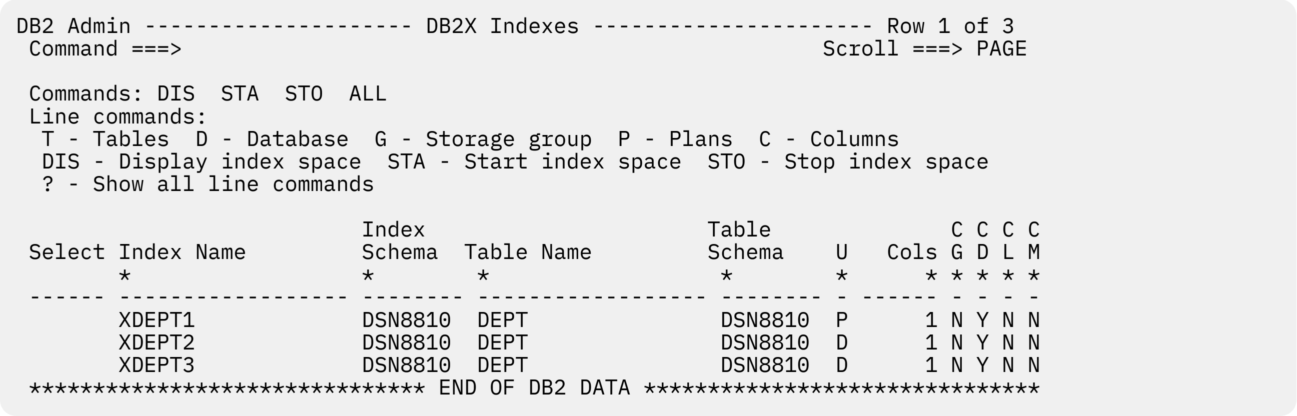
```
 DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

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

 Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C

 ----- ------------------ -------- - -------- -------- ------ ----------- ---- -
       VDEPT              DSN81010 V DSN8D10A DSN8S10D      4          -1   0
       VHDEPT             DSN81010 V DSN8D10A DSN8S10D      5          -1   0
       VEMP               DSN81010 V DSN8D10A DSN8S10E      5          -1   0
       VPROJ              DSN81010 V DSN8D10A DSN8S10P      8          -1   0
       VACT               DSN81010 V DSN8D10A DSN8S10P      3          -1   0
       VPROJACT           DSN81010 V DSN8D10A DSN8S10P      5           1   0
       VEMPPROJACT        DSN81010 V DSN8D10A DSN8S10P      6          -1   0
       VCONA              DSN81010 V DSN8D10P DSN8S10C      5          -1   0
       VOPTVAL            DSN81010 V DSN8D10P DSN8S10C     11          -1   0
       VDSPTXT            DSN81010 V DSN8D10P DSN8S10C      3          -1   0
       VDEPMG1            DSN81010 V DSN8D10A DSN8S10D      7          -1   0
       VEMPDPT1           DSN81010 V DSN8D10A DSN8S10D      7          -1   0
```

*Figure 63. **Tables, Views, and Aliases (ADB21T)** panel – displaying views*

Option X. Indexes

The **Indexes (ADB21X)** panel displays the indexes in the Db2 catalog.

Select option X on the **System Catalog (ADB21)** panel to display the **Indexes (ADB21X)** panel, as shown in the following figure.



```
DB2 Admin --------------------- DB2X Indexes ---------------------- Row 1 of 3
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  STA  STO  ALL
 Line commands:
  T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
  DIS - Display index space  STA - Start index space  STO - Stop index space
  ? - Show all line commands

                           Index                      Table              C C C C
 Select Index Name         Schema  Table Name         Schema    U   Cols G D L M
        *                  *        *                  *        *      * * * * *
 ------ ------------------ -------- ------------------ -------- - ------ - - - -
        XDEPT1             DSN8810  DEPT               DSN8810  P      1 N Y N N
        XDEPT2             DSN8810  DEPT               DSN8810  D      1 N Y N N
        XDEPT3             DSN8810  DEPT               DSN8810  D      1 N Y N N
 ******************************* END OF DB2 DATA *******************************
```

*Figure 64. **Indexes (ADB21X)** panel*

For descriptions of any of the columns on this panel, see the online help (PF1).

On the **Indexes (ADB21X)** panel, you can issue many line commands. Enter a question mark (?) on a row to view all valid line commands. These line commands include:

C
Displays columns in the index.

CON
Displays tables constrained by the index.

DIS
Displays index space status.

DP
Displays dependencies.

The following primary commands are valid on this panel:

ALL
Lists all objects of a specified type for each object in a list of objects. Supported values for this panel are:

ALL T
Shows all tables associated with the listed indexes.

DIS
Issues a Db2 DISPLAY command on multiple indexes.

STA
Issues a Db2 START command on multiple indexes.

STO
Issues a Db2 STOP command on multiple indexes.

UT or UTL or UTIL
Generates utility JCL for all listed indexes.

XSPACE
Displays the index space names.

If the size of the statements generated by the DIS, STA, or STO primary command exceeds 32K (an ISPF limit) or the number of statements generated exceeds 10, you will be prompted to send the statements to a batch job or a work statement list (WSL).

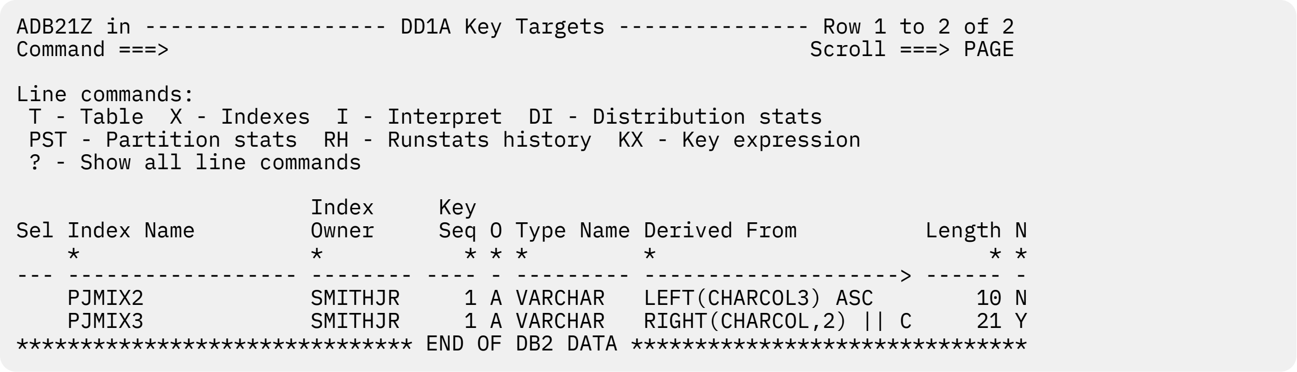
XML indexes

XML indexes use the same Db2 catalog support structure as extended indexes (indexes on expressions.)

- Panel ADB21X supports the extended indexes and columns in SYSINDEXES and SYSINDEXPART.
- The KT line command on panel ADB21X displays the information from SYSKEYTARGETS on panel (ADB21Z).
- Line commands are available to display statistics for catalog tables SYSKEYTARGET* and SYSKEYTGT* in the same way as SYSCOL* statistics tables.
- The XC line command on panel ADB21T supports extended indexes.

The following panels support extended indexes and columns in SYSINDEXES and SYSINDEXPART:

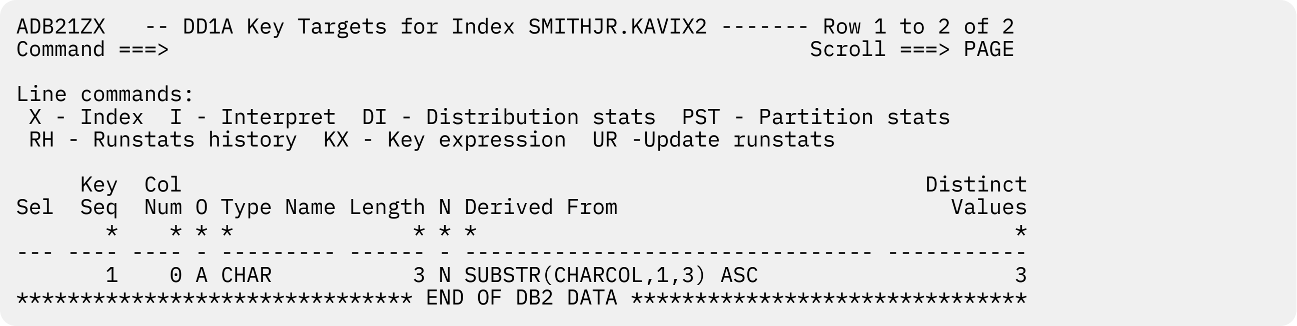
ADB21Z - Key Targets
Lists the key targets that participate in an extended index definition. Display ADB21Z by issuing the line command KT – Key Targets against a table entry on panel ADB21T.



```
ADB21Z in ------------------- DD1A Key Targets --------------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

Line commands:
 T - Table  X - Indexes  I - Interpret  DI - Distribution stats
 PST - Partition stats  RH - Runstats history  KX - Key expression
 ? - Show all line commands

                       Index     Key
Sel Index Name         Owner     Seq O Type Name Derived From          Length N
    *                  *           * * *         *                          * *
--- ------------------ -------- ---- - --------- --------------------> ------ -
    PJMIX2             SMITHJR     1 A VARCHAR   LEFT(CHARCOL3) ASC        10 N
    PJMIX3             SMITHJR     1 A VARCHAR   RIGHT(CHARCOL,2) || C     21 Y
******************************* END OF DB2 DATA *******************************
```

Figure 65. Key targets panel (ADB21Z)

ADB21ZX - Key Targets for Index
Lists the key targets that participate in an extended index definition for each of the extended indexes of a table. Display ADB21ZX by issuing the line command 'KT – Key Targets" against an index on panel ADB21X.



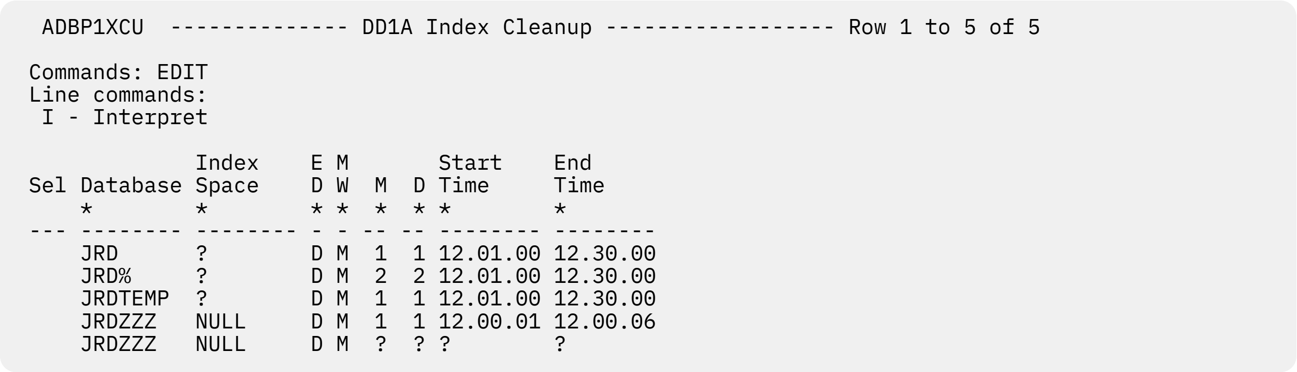
```
ADB21ZX   -- DD1A Key Targets for Index SMITHJR.KAVIX2 ------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

Line commands:
 X - Index  I - Interpret  DI - Distribution stats  PST - Partition stats
 RH - Runstats history  KX - Key expression  UR -Update runstats

     Key  Col                                                          Distinct
Sel  Seq  Num O Type Name Length N Derived From                          Values
       *    * * *             * * *                                          *
--- ---- ---- - --------- ------ - -------------------------------- -----------
       1    0 A CHAR           3 N SUBSTR(CHARCOL,1,3) ASC                    3
******************************* END OF DB2 DATA *******************************
```

Figure 66. Key targets for index panel (ADB21ZX)

Option XCU. Index Cleanup

Use the **Index Cleanup** panel to display information about index cleanup activities in the Db2 catalog.

Select option XCU on the **System Catalog** panel to display the **Index Cleanup** panel, as shown in the following figure.



```
 ADBP1XCU  -------------- DD1A Index Cleanup ------------------ Row 1 to 5 of 5

Commands: EDIT
Line commands:
 I - Interpret

             Index    E M       Start    End
Sel Database Space    D W  M  D Time     Time
    *        *        * *  *  * *        *
--- -------- -------- - - -- -- -------- --------
    JRD      ?        D M  1  1 12.01.00 12.30.00
    JRD%     ?        D M  2  2 12.01.00 12.30.00
    JRDTEMP  ?        D M  1  1 12.01.00 12.30.00
    JRDZZZ   NULL     D M  1  1 12.00.01 12.00.06
    JRDZZZ   NULL     D M  ?  ? ?        ?
```

Figure 67. Index Cleanup panel (ADBP1XCU)

The following primary command is valid on this panel:

EDIT
Enables edit of the index cleanup entries. You can delete, insert, or modify entries without having to use Db2 data manipulation language (DML).

The following line command is valid on this panel:

I
Provides information about the state object and timestamp information about the object cleanup.

The following columns are on this panel:

Database
Name of the database that contains the index.

Index Space
Name of the index space.

ED
Enable or disable. It specifies whether the row enables or disables cleanup for the specified index space.

MW
Month and week. It indicates how the value of the DAY is interpreted:

M
The value of the DAY column is interpreted as a day of the month.

W
The value of the DAY column is interpreted as a day of the week.

M
Indicates the month in which the time window applies. If this column contains NULL, the time window applies to all months.

D
Indicates the day of the month or the week. Indicates the day of the month, if M is specified in the MW column. Indicates day of the week if W is specified in the MW column, or if the MW column is null. When this column represents the day of the week, 1 is for Monday and 7 is Sunday. If this column contains NULL, the time window applies to every day of the month or to every day of the week.

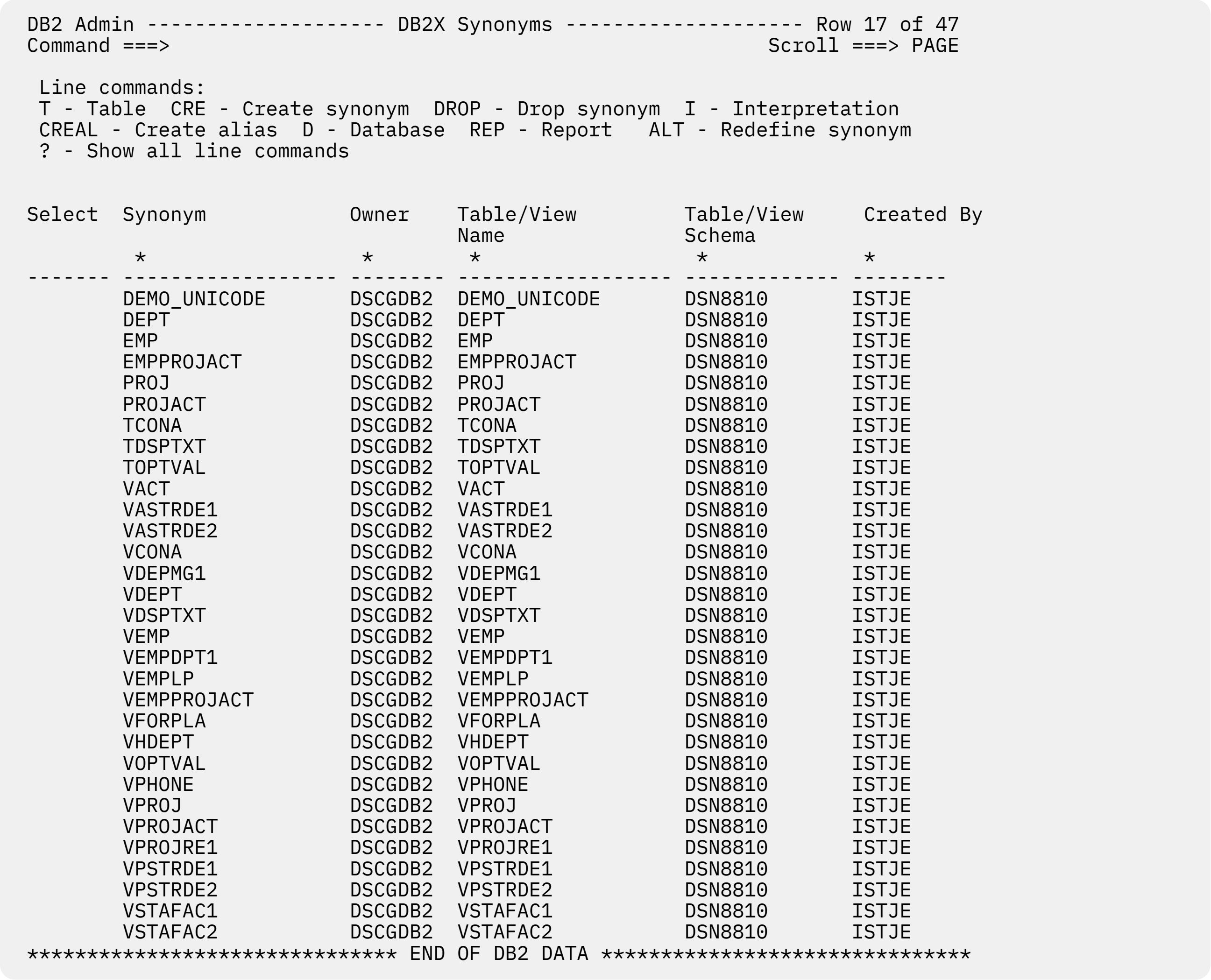
Start Time
The time of the day at which the row starts to apply cleanup. If this column contains a null value, the row applies cleanup at all times on the specified day.

End Time
The time of the day at which the row ends to apply cleanup. If this column contains a null value, the row applies cleanup at all times on the specified day.

Option Y. Synonyms

The **Synonyms** panel displays the synonyms in the Db2 catalog.

Select option Y on the **System Catalog** panel to display the **Synonyms** panel, as shown in the following figure.



```
DB2 Admin -------------------- DB2X Synonyms -------------------- Row 17 of 47
Command ===>                                                  Scroll ===> PAGE

 Line commands:
 T - Table  CRE - Create synonym  DROP - Drop synonym  I - Interpretation
 CREAL - Create alias  D - Database  REP - Report   ALT - Redefine synonym
 ? - Show all line commands


Select  Synonym            Owner    Table/View         Table/View     Created By
                                    Name               Schema
         *                  *        *                  *              *
------- ------------------ -------- ------------------ ------------- --------
        DEMO_UNICODE       DSCGDB2  DEMO_UNICODE       DSN8810       ISTJE
        DEPT               DSCGDB2  DEPT               DSN8810       ISTJE
        EMP                DSCGDB2  EMP                DSN8810       ISTJE
        EMPPROJACT         DSCGDB2  EMPPROJACT         DSN8810       ISTJE
        PROJ               DSCGDB2  PROJ               DSN8810       ISTJE
        PROJACT            DSCGDB2  PROJACT            DSN8810       ISTJE
        TCONA              DSCGDB2  TCONA              DSN8810       ISTJE
        TDSPTXT            DSCGDB2  TDSPTXT            DSN8810       ISTJE
        TOPTVAL            DSCGDB2  TOPTVAL            DSN8810       ISTJE
        VACT               DSCGDB2  VACT               DSN8810       ISTJE
        VASTRDE1           DSCGDB2  VASTRDE1           DSN8810       ISTJE
        VASTRDE2           DSCGDB2  VASTRDE2           DSN8810       ISTJE
        VCONA              DSCGDB2  VCONA              DSN8810       ISTJE
        VDEPMG1            DSCGDB2  VDEPMG1            DSN8810       ISTJE
        VDEPT              DSCGDB2  VDEPT              DSN8810       ISTJE
        VDSPTXT            DSCGDB2  VDSPTXT            DSN8810       ISTJE
        VEMP               DSCGDB2  VEMP               DSN8810       ISTJE
        VEMPDPT1           DSCGDB2  VEMPDPT1           DSN8810       ISTJE
        VEMPLP             DSCGDB2  VEMPLP             DSN8810       ISTJE
        VEMPPROJACT        DSCGDB2  VEMPPROJACT        DSN8810       ISTJE
        VFORPLA            DSCGDB2  VFORPLA            DSN8810       ISTJE
        VHDEPT             DSCGDB2  VHDEPT             DSN8810       ISTJE
        VOPTVAL            DSCGDB2  VOPTVAL            DSN8810       ISTJE
        VPHONE             DSCGDB2  VPHONE             DSN8810       ISTJE
        VPROJ              DSCGDB2  VPROJ              DSN8810       ISTJE
        VPROJACT           DSCGDB2  VPROJACT           DSN8810       ISTJE
        VPROJRE1           DSCGDB2  VPROJRE1           DSN8810       ISTJE
        VPSTRDE1           DSCGDB2  VPSTRDE1           DSN8810       ISTJE
        VPSTRDE2           DSCGDB2  VPSTRDE2           DSN8810       ISTJE
        VSTAFAC1           DSCGDB2  VSTAFAC1           DSN8810       ISTJE
        VSTAFAC2           DSCGDB2  VSTAFAC2           DSN8810       ISTJE
******************************* END OF DB2 DATA *******************************
```

Figure 68. Synonyms panel (ADB21Y)

The following columns are on this panel:

Select
Input field where you enter one of the line commands listed on the panel.

Synonym
Synonym for the table or view.

Owner
Authorization ID of the owner of the synonym.

Table/View Name
Name of the table or view.

Table/View Schema
The schema of the table or view.

Created By
Primary authorization ID of the user who created the synonym.

Option AO. Authorization options

You can use the **System Catalog (ADB21)** panel - Authorization options to manage authorizations for objects in the Db2 catalog.

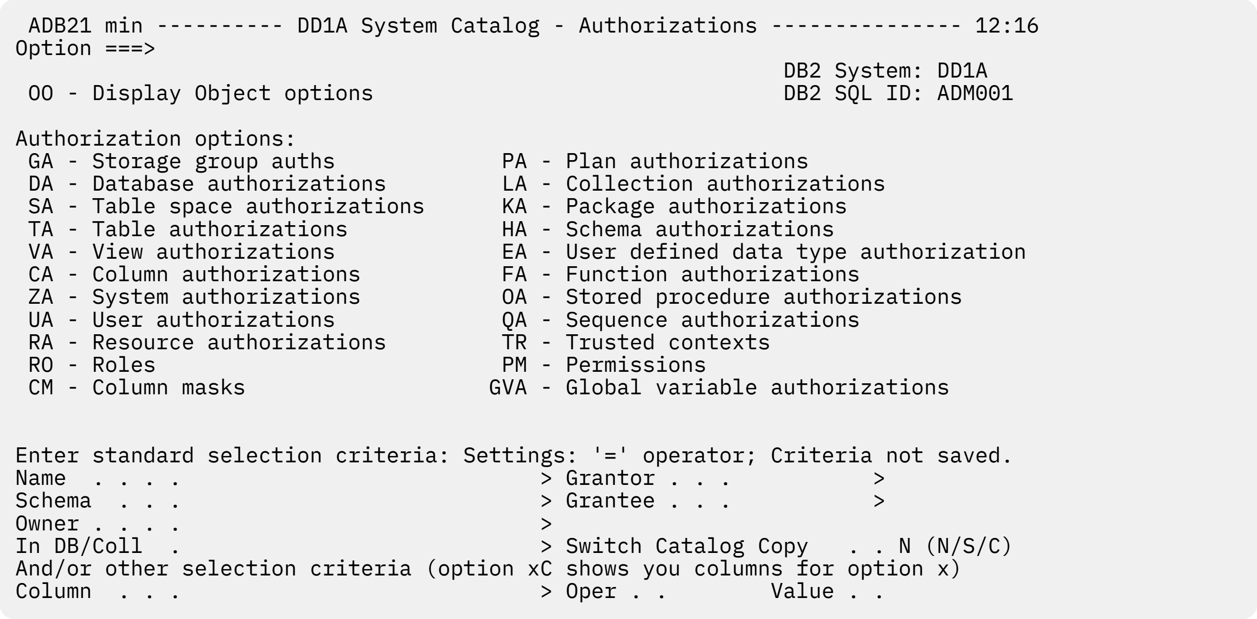
You can manage authorizations for the following database objects:

- Collections
- Columns
- Databases
- Data types
- Functions
- Global variables
- Packages
- Plans
- Resources
- Schemas
- Sequences
- Storage groups
- Stored procedures
- System privileges
- Table spaces
- Tables
- Users
- User-defined data types
- Views

To display the authorizations granted on a particular type of database object:

1. On the **System Catalog (ADB21)** panel, specify option A0 and press Enter.

   The **System Catalog (ADB21)** panel - Authorization options are displayed:



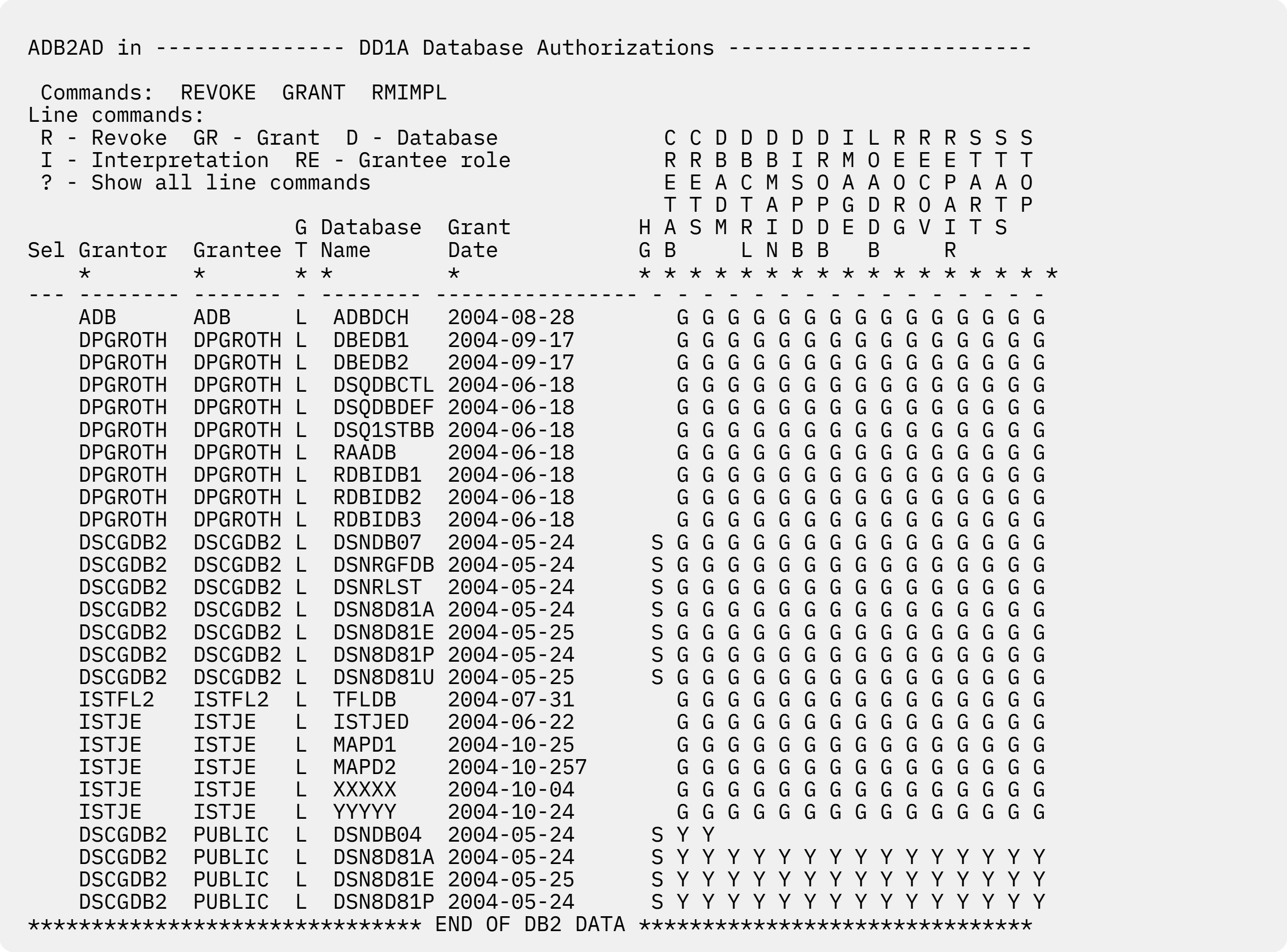
```
 ADB21 min ---------- DD1A System Catalog - Authorizations --------------- 12:16
Option ===>
                                                            DB2 System: DD1A
 OO - Display Object options                                DB2 SQL ID: ADM001

Authorization options:
 GA - Storage group auths             PA - Plan authorizations
 DA - Database authorizations         LA - Collection authorizations
 SA - Table space authorizations      KA - Package authorizations
 TA - Table authorizations            HA - Schema authorizations
 VA - View authorizations             EA - User defined data type authorization
 CA - Column authorizations           FA - Function authorizations
 ZA - System authorizations           OA - Stored procedure authorizations
 UA - User authorizations             QA - Sequence authorizations
 RA - Resource authorizations         TR - Trusted contexts
 RO - Roles                           PM - Permissions
 CM - Column masks                   GVA - Global variable authorizations


Enter standard selection criteria: Settings: '=' operator; Criteria not saved.
Name  . . . .                            > Grantor . . .           >
Schema  . . .                            > Grantee . . .           >
Owner . . . .                            >
In DB/Coll  .                            > Switch Catalog Copy   . . N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .                            > Oper . .        Value . .
```

*Figure 69. **System Catalog (ADB21)** panel - Authorization options*

2. Type the two-character option that corresponds to the type of object in the **Option** field.
3. Optionally, specify a value in either the **Grantor** or **Grantee** fields.

   Recommendation: For optimum performance when using any authorization option (xA), specify a value in either the **Grantor** or **Grantee** fields of the **System Catalog (ADB21)** panel - Authorization options panel.
4. Press Enter.

For example, to display authorization information for databases, type DA in the **Option** field, and press Enter. The **Database Authorizations (ADB2AD)** panel is displayed:



```
ADB2AD in --------------- DD1A Database Authorizations ------------------------

 Commands:  REVOKE  GRANT  RMIMPL
Line commands:
 R - Revoke  GR - Grant  D - Database             C C D D D D D I L R R R S S S
 I - Interpretation  RE - Grantee role            R R B B B I R M O E E E T T T
 ? - Show all line commands                       E E A C M S O A A O C P A A O
                                                  T T D T A P P G D R O A R T P
                     G Database  Grant          H A S M R I D D E D G V I T S
Sel Grantor  Grantee T Name      Date           G B     L N B B   B     R
    *        *       * *         *              * * * * * * * * * * * * * * * * *
--- -------- ------- - -------- ---------------- - - - - - - - - - - - - - - - -
    ADB      ADB     L  ADBDCH   2004-08-28        G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  DBEDB1   2004-09-17        G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  DBEDB2   2004-09-17        G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  DSQDBCTL 2004-06-18        G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  DSQDBDEF 2004-06-18        G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  DSQ1STBB 2004-06-18        G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  RAADB    2004-06-18        G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  RDBIDB1  2004-06-18        G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  RDBIDB2  2004-06-18        G G G G G G G G G G G G G G G
    DPGROTH  DPGROTH L  RDBIDB3  2004-06-18        G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSNDB07  2004-05-24      S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSNRGFDB 2004-05-24      S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSNRLST  2004-05-24      S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSN8D81A 2004-05-24      S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSN8D81E 2004-05-25      S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSN8D81P 2004-05-24      S G G G G G G G G G G G G G G G
    DSCGDB2  DSCGDB2 L  DSN8D81U 2004-05-25      S G G G G G G G G G G G G G G G
    ISTFL2   ISTFL2  L  TFLDB    2004-07-31        G G G G G G G G G G G G G G G
    ISTJE    ISTJE   L  ISTJED   2004-06-22        G G G G G G G G G G G G G G G
    ISTJE    ISTJE   L  MAPD1    2004-10-25        G G G G G G G G G G G G G G G
    ISTJE    ISTJE   L  MAPD2    2004-10-257       G G G G G G G G G G G G G G G
    ISTJE    ISTJE   L  XXXXX    2004-10-04        G G G G G G G G G G G G G G G
    ISTJE    ISTJE   L  YYYYY    2004-10-24        G G G G G G G G G G G G G G G
    DSCGDB2  PUBLIC  L  DSNDB04  2004-05-24      S Y Y
    DSCGDB2  PUBLIC  L  DSN8D81A 2004-05-24      S Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
    DSCGDB2  PUBLIC  L  DSN8D81E 2004-05-25      S Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
    DSCGDB2  PUBLIC  L  DSN8D81P 2004-05-24      S Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
******************************* END OF DB2 DATA *******************************
```

*Figure 70. **Database Authorizations (ADB2AD)** panel*

All of the authorization panels are structured similarly to the **Database Authorizations (ADB2AD)** panel; they list the detailed authorization information about the type of database object that you selected. To limit the display to only explicitly granted authorizations, use the RMIMPL (remove implicit) command.

From the authorization panels, you can grant and revoke authorizations for a particular object or for all the objects that are displayed.

Types of panels

Db2 Admin Tool uses table display, BROWSE, VIEW, and SQL error-display panels.

Table display panels

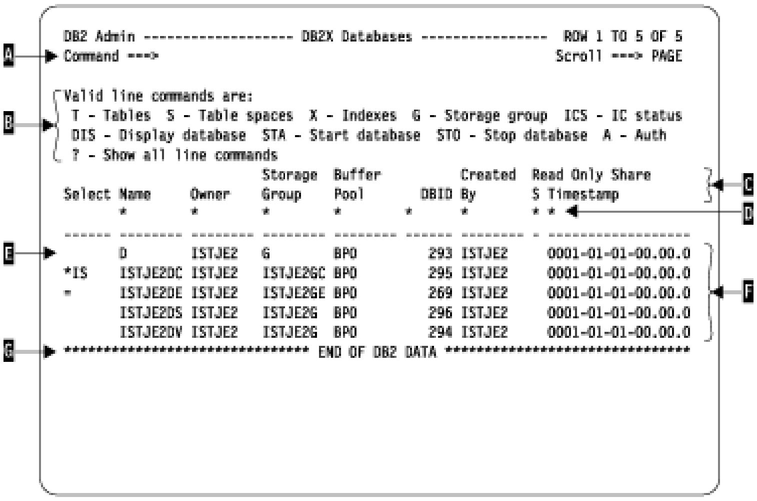
Table display panels contain ISPF tables that show information about Db2 objects. From these panels, you can run various Db2 Admin Tool functions.

Important: In some cases, fields on Db2 Admin Tool panels might be hidden or output-only, as follows:

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

For example, if the version of Db2 that is used has a restriction related to a field, that field might be hidden or output-only. Hidden fields can cause blank lines or spaces; disregard them.

The panel in the following figure shows the areas on a typical table display panel.



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

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

Figure 71. Table Display panel layout

First row of the panel
Contains the Db2 Admin Tool panel name and a count of data rows. The row count reflects an initial search done by your Db2 subsystem.

A Command line
On this line you can enter any Db2 command, ISPF command, or Db2 Admin Tool primary command.

B Line command description area
Indicates the Db2 Admin Tool line commands that you can issue from a particular table display panel. You issue a line command in the Select field (area E). When the panel does not have not enough space to list all valid line commands, only the most frequently used line commands are shown. To display all other valid line commands, specify a question mark (?) in the Select field, and press Enter.

C Column headers
Contains the names of the columns that have data.

D Search arguments
Use this area to specify search criteria for the data that is displayed in the panel. ISPF generic search argument rules apply. For columns that contain alphabetic characters, the asterisk (*) under the column name marks the beginning (left-justified) of the area in which you can enter search criteria to limit the information that Db2 Admin Tool returns. For columns that contain numeric characters, the

asterisk (*) marks the end (right-justified) of the area. For example, you can enter D050 in the Name column to display only those databases whose names begin with D050.

Specify a blank to reset the filter for the column. For advanced column filtering, use the SEARCH (or SARG) command.

E Select column

Use the Select column to issue Db2 Admin Tool line commands (shown in area B) against Db2 objects that are listed in the table display panel.

If you specify a line command or update a row in the table display and also issue a scroll request by pressing PF7 or PF8, the line command or row update is processed, and the scroll request is ignored.

F Rows returned

Shows the rows that Db2 returns based on the options that you selected, the commands that you issued, or the search criteria that you entered. For example, to display the panel shown in the previous figure, on the **System Catalog (ADB21)** panel, request that all databases owned by ISTJE2 be displayed.

G End of data marker

Indicates the end of the data returned from Db2.

BROWSE and VIEW panels

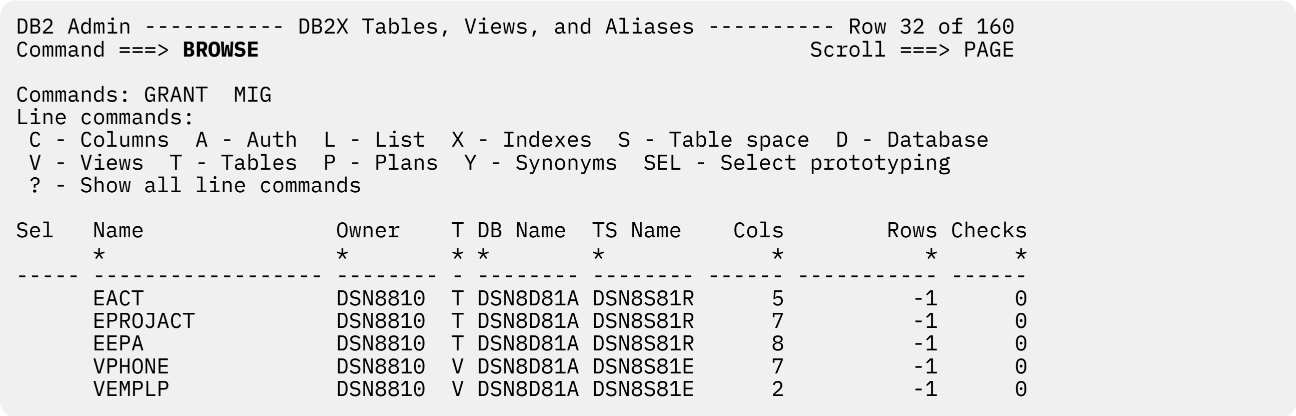
BROWSE and VIEW panels contain details about Db2 objects.

From any table display panel, you can issue BROWSE and VIEW commands as follows:

- The BROWSE (or BR) primary command and the BR line command open an ISPF BROWSE session. In a BROWSE session, you cannot edit the data.
- The VIEW (or VI) primary command and the VI line command open an ISPF VIEW session. In a VIEW session, you can edit the data in a data set by using ISPF commands.

For both BROWSE and VIEW, the primary commands display information about the entire panel, and the line commands display information about only the selected object.

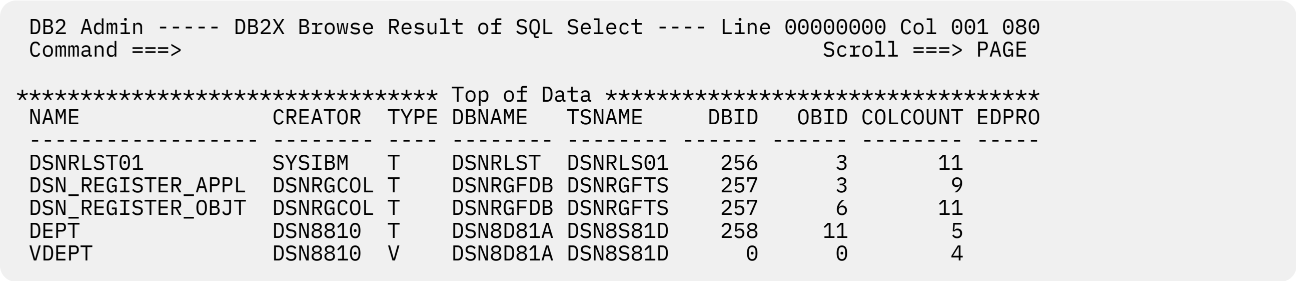
For example, on the following table display panel, the **Tables, Views, and Aliases (ADB21T)** panel, suppose that you issue the BROWSE primary command:



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

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

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

*Figure 72. BROWSE command on the **Tables, Views, and Aliases (ADB21T)** panel*

The following output is generated in an ISPF BROWSE session. The first line is a header with column names from the Db2 catalog. To display the remaining columns, scroll to the right.



```
 DB2 Admin ----- DB2X Browse Result of SQL Select ---- Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

********************************* Top of Data **********************************
 NAME               CREATOR  TYPE DBNAME   TSNAME     DBID   OBID COLCOUNT EDPRO
 ------------------ -------- ---- -------- -------- ------ ------ -------- -----
 DSNRLST01          SYSIBM   T    DSNRLST  DSNRLS01    256      3       11
 DSN_REGISTER_APPL  DSNRGCOL T    DSNRGFDB DSNRGFTS    257      3        9
 DSN_REGISTER_OBJT  DSNRGCOL T    DSNRGFDB DSNRGFTS    257      6       11
 DEPT               DSN8810  T    DSN8D81A DSN8S81D    258     11        5
 VDEPT              DSN8810  V    DSN8D81A DSN8S81D      0      0        4
```

Figure 73. Output from the BROWSE command

Tip: When you use BROWSE or VIEW, all columns for that Db2 catalog table are displayed, even those columns that are considered "not used" by Db2. [To determine which columns are "not used," see the documentation for the catalog table in Db2 catalog tables (Db2 12 for z/OS).] If you do not want these columns displayed, use the I (Interpret) line command on the table panel instead of the BROWSE or VIEW command.

Db2 Admin Tool can also display data in tables that contain binary large objects (BLOBs), character large objects (CLOBs), double-byte character large objects (DBCLOBs), and ROWID columns:

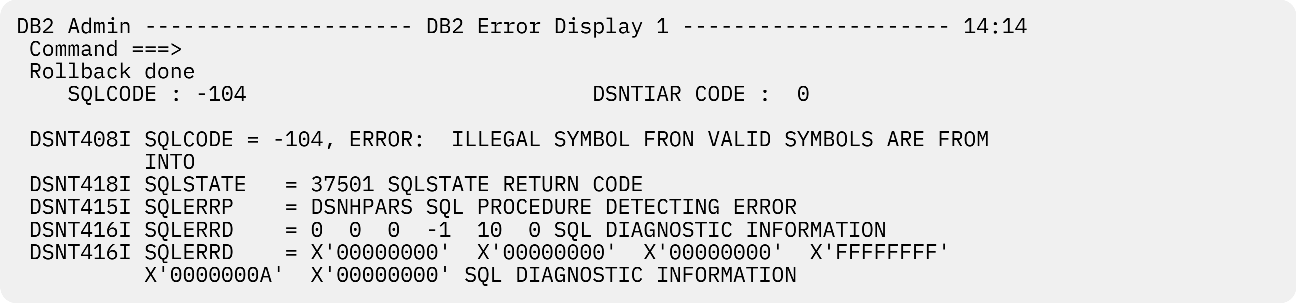
- For BLOBs, Db2 Admin Tool retrieves up to 128 bytes per column and displays the data in hexadecimal format.
- For CLOBs, Db2 Admin Tool retrieves up to 256 bytes per column and displays the data in character format.
- For DBCLOBs, Db2 Admin Tool retrieves up to 128 bytes per column and displays the data in hexadecimal format.
- ROWIDs are displayed in hexadecimal format.

SQL error-display panels

If an error occurs when running an SQL statement, Db2 Admin Tool displays the SQL code and error message on a separate panel called an *SQL error-display panel*.

To correct the SQL statement, press END, which redisplays the panel where you originally issued the SQL statement. Db2 Admin Tool positions the cursor at the point in the SQL statement where Db2 found the error.

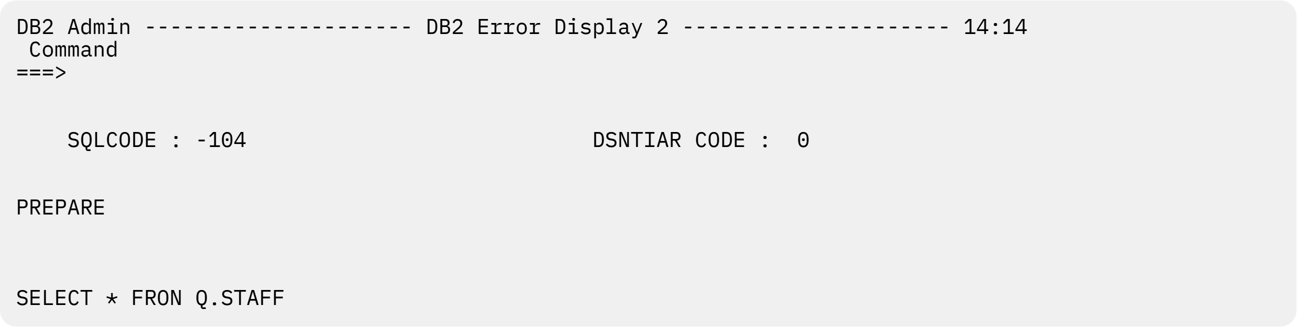
The following figure shows the error panel that Db2 Admin Tool displays when the following SQL statement (containing a spelling error) is issued: `SELECT * FRON Q.STAFF`.



```
DB2 Admin -------------------- DB2 Error Display 1 -------------------- 14:14
 Command ===>
 Rollback done
    SQLCODE : -104                           DSNTIAR CODE :  0

 DSNT408I SQLCODE = -104, ERROR:  ILLEGAL SYMBOL FRON VALID SYMBOLS ARE FROM
          INTO
 DSNT418I SQLSTATE   = 37501 SQLSTATE RETURN CODE
 DSNT415I SQLERRP    = DSNHPARS SQL PROCEDURE DETECTING ERROR
 DSNT416I SQLERRD    = 0  0  0  -1  10  0 SQL DIAGNOSTIC INFORMATION
 DSNT416I SQLERRD    = X'00000000'  X'00000000'  X'00000000'  X'FFFFFFFF'
          X'0000000A'  X'00000000' SQL DIAGNOSTIC INFORMATION
```

Figure 74. Error Display panel (part 1 of 2)

Press Enter to see error panel two, as shown in the following figure.



```
DB2 Admin -------------------- DB2 Error Display 2 -------------------- 14:14
 Command
===>


    SQLCODE : -104                          DSNTIAR CODE :  0


PREPARE



SELECT * FRON Q.STAFF
```

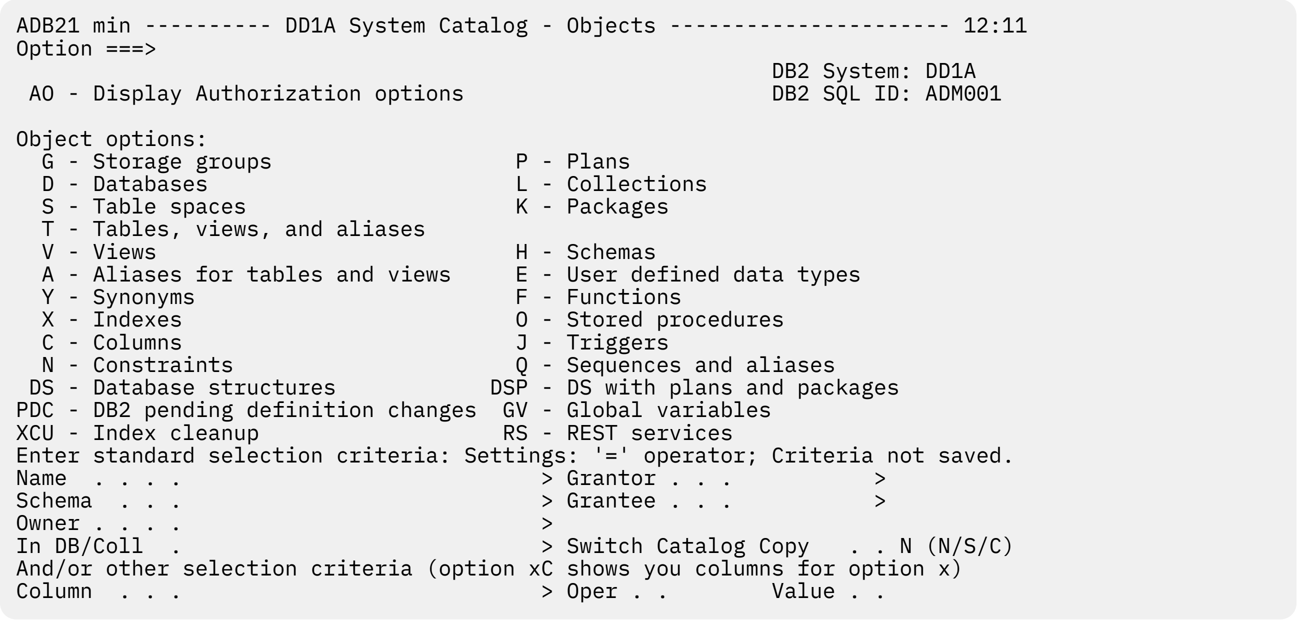
Figure 75. Error Display panel (part 2 of 2)

Press END to redisplay the panel on which you entered the incorrect SQL statement.

Panel IDs

A *panel ID* identifies a panel. When turned on, the panel IDs are displayed in the upper left corner of the panels.

For example, on the following panel, the panel ID is ADB21:



```
ADB21 min ---------- DD1A System Catalog - Objects --------------------- 12:11
Option ===>
                                                           DB2 System: DD1A
 AO - Display Authorization options                        DB2 SQL ID: ADM001

Object options:
  G - Storage groups                    P - Plans
  D - Databases                         L - Collections
  S - Table spaces                      K - Packages
  T - Tables, views, and aliases
  V - Views                             H - Schemas
  A - Aliases for tables and views      E - User defined data types
  Y - Synonyms                          F - Functions
  X - Indexes                           O - Stored procedures
  C - Columns                           J - Triggers
  N - Constraints                       Q - Sequences and aliases
 DS - Database structures             DSP - DS with plans and packages
PDC - DB2 pending definition changes   GV - Global variables
XCU - Index cleanup                    RS - REST services
Enter standard selection criteria: Settings: '=' operator; Criteria not saved.
Name  . . . .                         > Grantor . . .           >
Schema  . . .                         > Grantee . . .           >
Owner . . . .                         >
In DB/Coll  .                         > Switch Catalog Copy   . . N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .                         > Oper . .        Value . .
```

Figure 76. ***System Catalog (ADB21)*** *panel*

If panel IDs are not already displayed, you can turn them on by issuing the ISPF command `PANELID ON`.

If you are developing Db2 Admin Tool applications, you can use the panel ID to quickly locate the source code for a specific panel.

Note: When Db2 Admin Tool panels are discussed in this information, the name of the panel is followed by the panel ID in parentheses. For example: **System Catalog (ADB21)** panel. In this example, the title of the panel is **System Catalog**, and the ID is ADB21.

Determining valid values for input fields

You can use the *Look Up function* to determine valid values that you can specify in certain input fields. By using the Look Up function, you can save keystrokes and avoid typing errors. You can also avoid backing out of the current panel in order to search for the correct object.

About this task

Only some fields support the Look Up function. Fields that support the Look Up function are denoted by a question mark (?) at the end of the field as shown in the following example field:

```
Table name . . . . .                    >  (? to look up, * for all tables)
```

You can specify the question mark (?) Look Up character to search possible values that you can enter in the fields.

Procedure

To determine valid values for an input field:

1. In a field that supports the Look Up function, specify a question mark (?) and optionally a qualifier, and press Enter.

   To use the Look Up function with a qualifier, specify the first few characters of a name followed by a question mark (?), and press Enter. To include all the results that contain the qualifier for which you are searching, include the percent sign (%) as a wildcard with the qualifier. For example, if you specify TS01?, the function finds all names that start with TS01. If you specify %TS01?, the function finds all names that contain TS01.

   A list of valid choices is displayed.
2. To select a value from this list, specify a plus sign (+) beside your choice.

   For input fields that support more than one value, you can select multiple objects from the list by entering a plus sign next to each object that you want to select and pressing End.

   Specify the plus sign (+) Look Up character only to select an object from the list that is returned by the Look Up function. If you specify the plus sign on a table that is not provided by the Look Up function, an invalid line command error message is returned.

Example of using the Look Up function

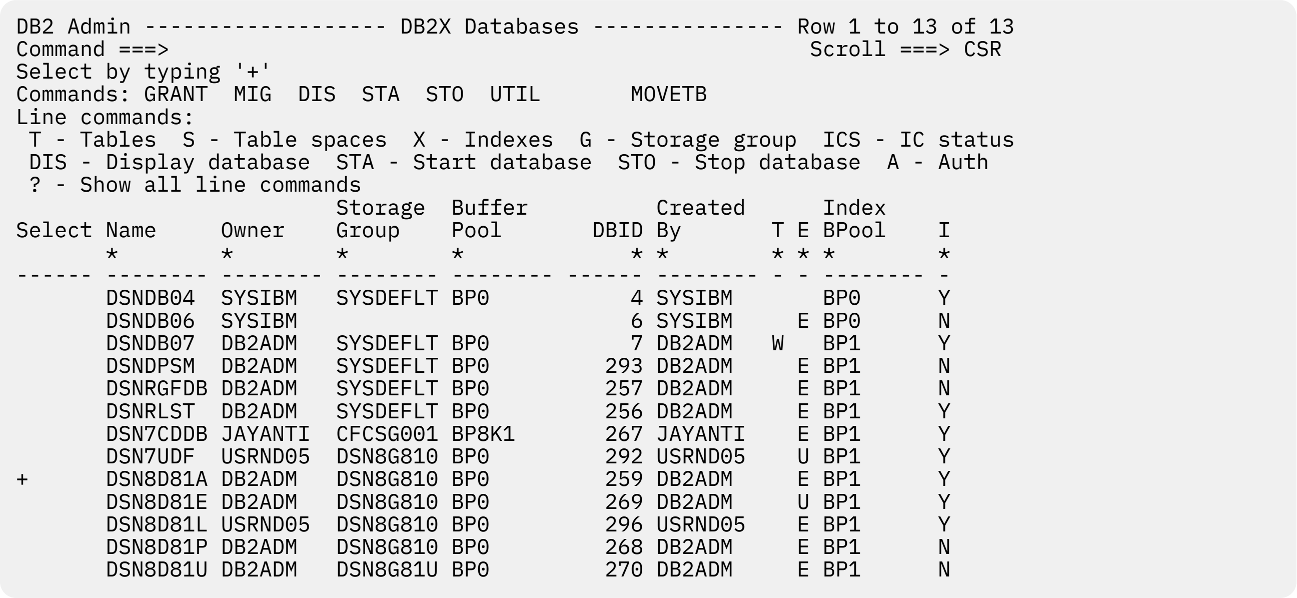
The following figures show an example of using the Db2 Admin Tool Look Up function. On this panel, the Db2 Admin Tool Look Up function is supported by two fields, TABLESPACE and IN, both of which show a question mark in the text to the right of the field. The table space name TSPACE01 is specified, but the character string DSN? is a request to display all databases that begin with DSN.

```
ADB26CS n -------------------DB2X Create Table Space ------------------- 06:28
Command ===> _________________________________________________________________

 CREATE

 TABLESPACE . . TSPACE01  (required table space name.  ? to look up)

 IN . . . . . . DSN?    (optional database. default=DSNDB04. ? to look up)

 Like:
 Database . . . ________ (optional existing database. ? to look up)
 Name . . . . . ________ (optional existing table space. ? to look up)
```

Figure 77. Using theDb2 Admin Tool Look Up function — requesting a Look Up on the Create Table Space panel (ADB26CS)

The following figure shows the results of using the Look Up function to display all databases that begin with DSN. Select an item by specifying a plus sign (+) in the Select field next to the desired table entry.

In this example, DSN8D81A is selected. When you press End, Db2 Admin Tool enters this name in the IN field of the previous panel.



```
DB2 Admin ------------------- DB2X Databases --------------- Row 1 to 13 of 13
Command ===>                                                  Scroll ===> CSR
Select by typing '+'
Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       *        *        *        *             * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       DSNDB04  SYSIBM   SYSDEFLT BP0           4 SYSIBM       BP0      Y
       DSNDB06  SYSIBM                          6 SYSIBM     E BP0      N
       DSNDB07  DB2ADM   SYSDEFLT BP0           7 DB2ADM   W   BP1      Y
       DSNDPSM  DB2ADM   SYSDEFLT BP0         293 DB2ADM     E BP1      N
       DSNRGFDB DB2ADM   SYSDEFLT BP0         257 DB2ADM     E BP1      N
       DSNRLST  DB2ADM   SYSDEFLT BP0         256 DB2ADM     E BP1      Y
       DSN7CDDB JAYANTI  CFCSG001 BP8K1       267 JAYANTI    E BP1      Y
       DSN7UDF  USRND05  DSN8G810 BP0         292 USRND05    U BP1      Y
+      DSN8D81A DB2ADM   DSN8G810 BP0         259 DB2ADM     E BP1      Y
       DSN8D81E DB2ADM   DSN8G810 BP0         269 DB2ADM     U BP1      Y
       DSN8D81L USRND05  DSN8G810 BP0         296 USRND05    E BP1      Y
       DSN8D81P DB2ADM   DSN8G810 BP0         268 DB2ADM     E BP1      N
       DSN8D81U DB2ADM   DSN8G81U BP0         270 DB2ADM     E BP1      N
```

*Figure 78. Using the DB2 Admin Look Up function — selecting an object on the **Databases (ADB21D)** panel*

Filtering data on panels

When you run queries to display information about Db2 objects or authorizations, you can filter the information that is displayed by using search arguments in certain input fields. After the query is run, you can also filter the resulting information that is displayed on the table display panel.

About this task

Depending on when you want to filter the information, use one of the following procedures:

- “Filtering data before running a query to display object or authorization information” on page 198
- “Filtering data on a panel after the query result is returned” on page 200

Filtering data before running a query to display object or authorization information

Procedure

- On the **System Catalog (ADB21)** panel, specify the filtering criteria at the bottom of the panel.

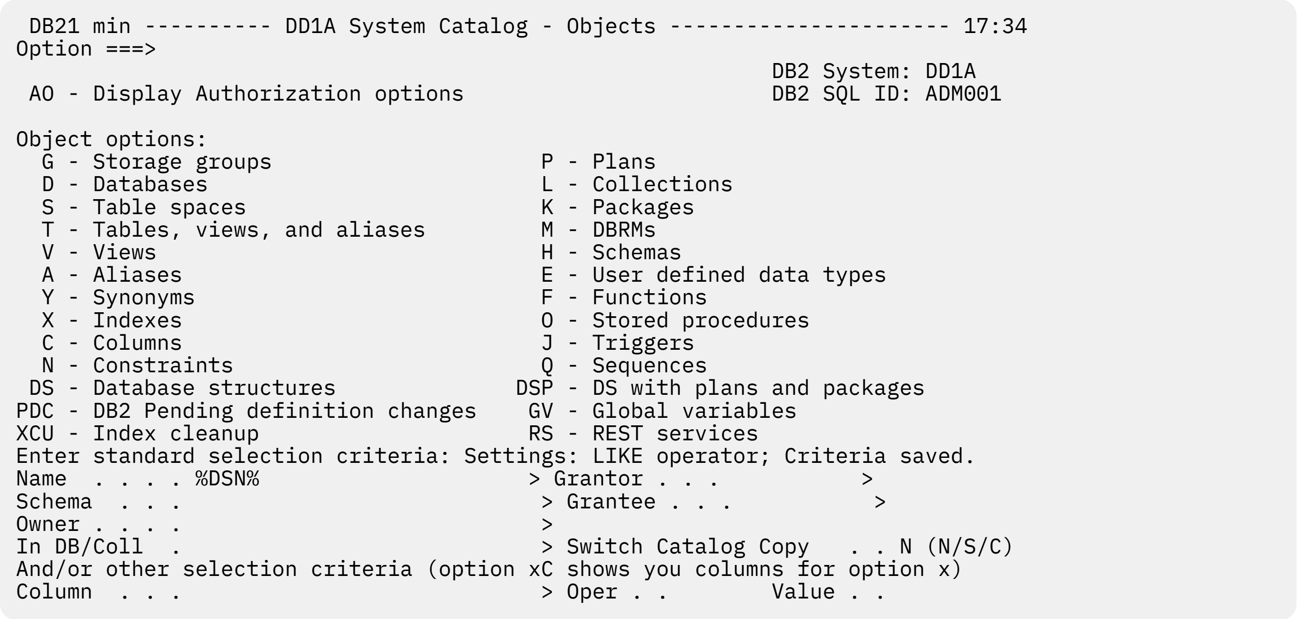
  Use the following guidance:

  – **Wildcard characters:** In your search argument, you can specify a percent sign (%) or an asterisk (*) as a wildcard character. If you use an asterisk as a wildcard character, Db2 Admin Tool translates it to a percent sign. The asterisk is displayed as a percent sign when the panel is re-displayed.

  – **Case:** Lowercase characters in the search argument for **Name**, **Owner**, **In DB/Coll**, **Grantor**, and **Grantee** are translated to uppercase characters unless you change the Db2 Admin Tool default setting. To change the default setting, navigate to the **Admin Defaults (ADB2P2)** panel and set the **Capitalize object names** field to NO. With this setting, lowercase characters are not translated to uppercase characters where Db2 rules allow the name to contain lowercase letters. Lowercase characters cannot be translated to uppercase characters in database names, table space names, plan names, and package names that are not for trigger packages

Example

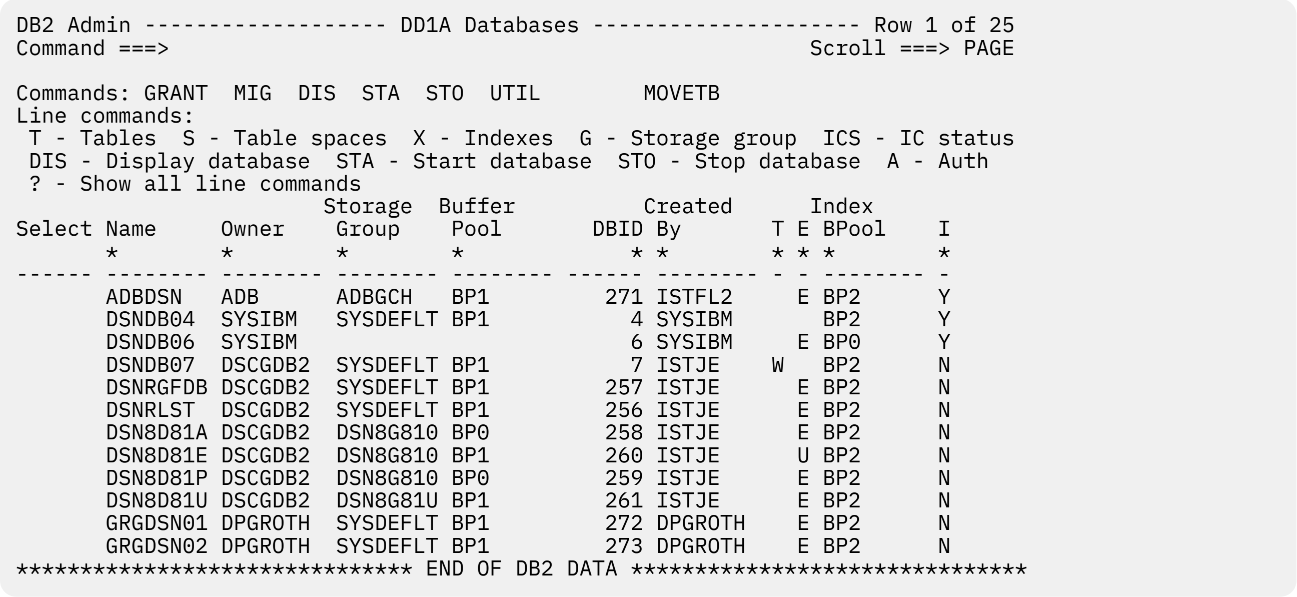
For example, the following panel shows how you can use a search argument with wildcard characters in the **Name** field on the **System Catalog (ADB21)** panel to display all the databases in the Db2 system catalog with names that contain the characters DSN.



```
 DB21 min ---------- DD1A System Catalog - Objects --------------------- 17:34
Option ===>
                                                           DB2 System: DD1A
 AO - Display Authorization options                        DB2 SQL ID: ADM001

Object options:
  G - Storage groups                     P - Plans
  D - Databases                          L - Collections
  S - Table spaces                       K - Packages
  T - Tables, views, and aliases         M - DBRMs
  V - Views                              H - Schemas
  A - Aliases                            E - User defined data types
  Y - Synonyms                           F - Functions
  X - Indexes                            O - Stored procedures
  C - Columns                            J - Triggers
  N - Constraints                        Q - Sequences
 DS - Database structures              DSP - DS with plans and packages
PDC - DB2 Pending definition changes    GV - Global variables
XCU - Index cleanup                     RS - REST services
Enter standard selection criteria: Settings: LIKE operator; Criteria saved.
Name  . . . . %DSN%                     > Grantor . . .           >
Schema  . . .                            > Grantee . . .           >
Owner . . . .                            >
In DB/Coll  .                            > Switch Catalog Copy   . . N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .                            > Oper . .        Value . .
```

*Figure 79. **System Catalog (ADB21)** panel - using search criteria*

When you press Enter, Db2 Admin Tool generates an SQL statement that searches the Db2 catalog using an SQL LIKE operator to qualify the search with the search criteria. The following figure shows the ISPF table display that Db2 Admin Tool returns. All databases that meet the search criteria (have a name that contains the characters 'DSN') are displayed.



```
DB2 Admin ------------------- DD1A Databases --------------------- Row 1 of 25
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL        MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                        Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       *        *        *        *            * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       ADBDSN   ADB      ADBGCH   BP1         271 ISTFL2     E BP2      Y
       DSNDB04  SYSIBM   SYSDEFLT BP1           4 SYSIBM       BP2      Y
       DSNDB06  SYSIBM                          6 SYSIBM     E BP0      Y
       DSNDB07  DSCGDB2  SYSDEFLT BP1           7 ISTJE    W   BP2      N
       DSNRGFDB DSCGDB2  SYSDEFLT BP1         257 ISTJE      E BP2      N
       DSNRLST  DSCGDB2  SYSDEFLT BP1         256 ISTJE      E BP2      N
       DSN8D81A DSCGDB2  DSN8G810 BP0         258 ISTJE      E BP2      N
       DSN8D81E DSCGDB2  DSN8G810 BP1         260 ISTJE      U BP2      N
       DSN8D81P DSCGDB2  DSN8G810 BP0         259 ISTJE      E BP2      N
       DSN8D81U DSCGDB2  DSN8G81U BP1         261 ISTJE      E BP2      N
       GRGDSN01 DPGROTH  SYSDEFLT BP1         272 DPGROTH    E BP2      N
       GRGDSN02 DPGROTH  SYSDEFLT BP1         273 DPGROTH    E BP2      N
******************************* END OF DB2 DATA *******************************
```

*Figure 80. **Databases (ADB21D)** panel - list of qualifying databases*

Filtering data on a panel after the query result is returned

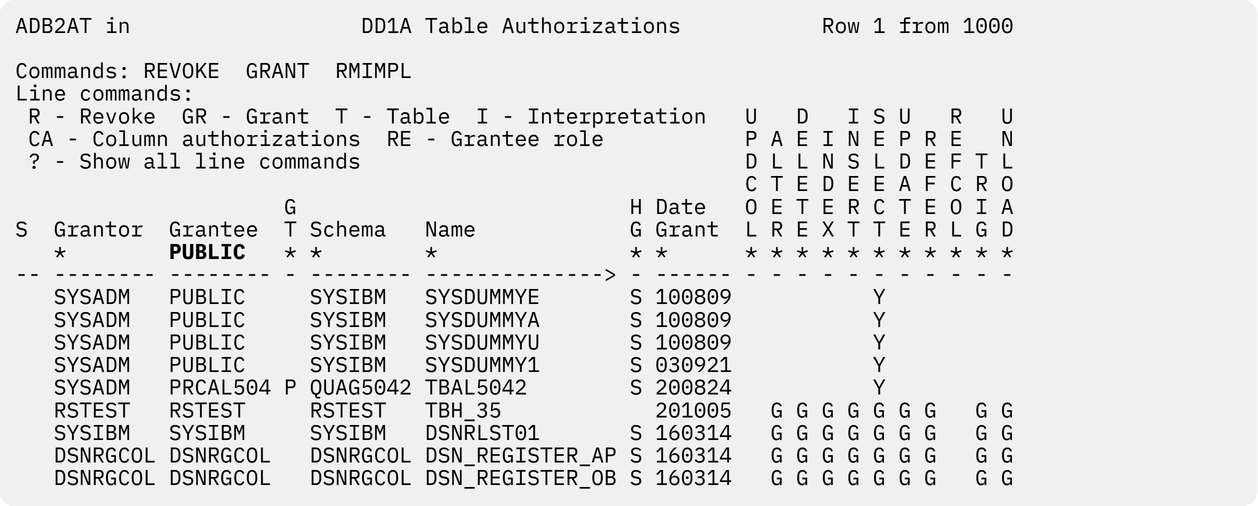
After Db2 Admin Tool returns the results of a query, you can filter the rows that are displayed on a table display panel.

Procedure

Complete one of the following actions:

- Use the search argument area on the panel to specify the filtering criteria.

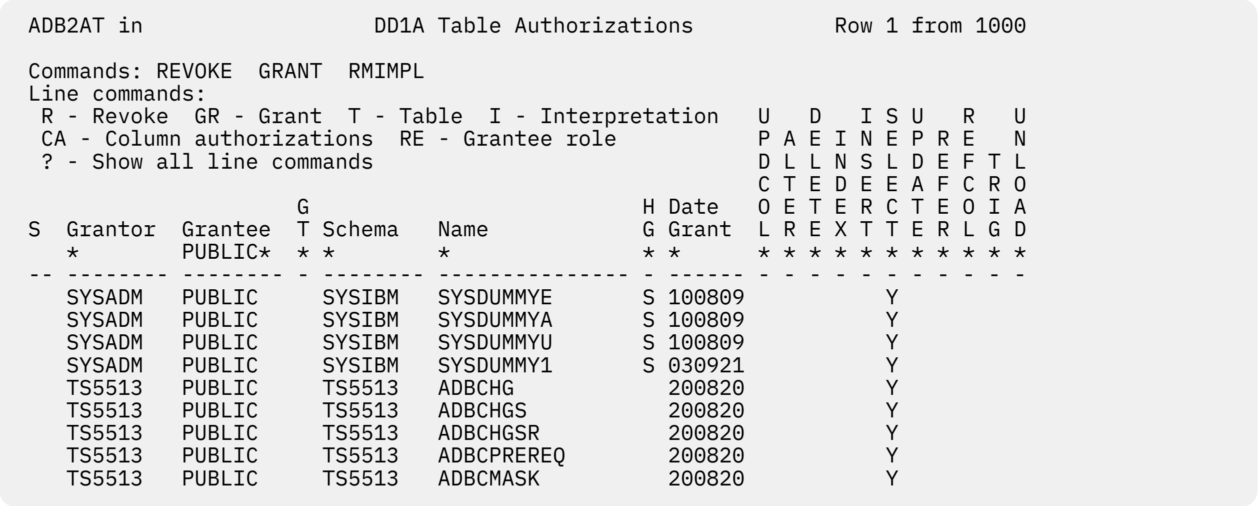
  For example, if you want to filter the following panel to see only those table authorizations that are granted to PUBLIC, you can type PUBLIC in the search argument area in the **Grantee** column:



```
ADB2AT in                  DD1A Table Authorizations           Row 1 from 1000

Commands: REVOKE  GRANT  RMIMPL
Line commands:
 R - Revoke  GR - Grant  T - Table  I - Interpretation   U   D   I S U   R   U
 CA - Column authorizations  RE - Grantee role           P A E I N E P R E   N
 ? - Show all line commands                              D L L N S L D E F T L
                                                         C T E D E E A F C R O
                     G                          H Date   O E T E R C T E O I A
S  Grantor  Grantee  T Schema   Name            G Grant  L R E X T T E R L G D
   *        PUBLIC   * *        *               * *      * * * * * * * * * * *
-- -------- -------- - -------- --------------> - ------ - - - - - - - - - - -
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYE       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYA       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYU       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMY1       S 030921           Y
   SYSADM   PRCAL504 P QUAG5042 TBAL5042        S 200824           Y
   RSTEST   RSTEST     RSTEST   TBH_35            201005   G G G G G G G   G G
   SYSIBM   SYSIBM     SYSIBM   DSNRLST01       S 160314   G G G G G G G   G G
   DSNRGCOL DSNRGCOL   DSNRGCOL DSN_REGISTER_AP S 160314   G G G G G G G   G G
   DSNRGCOL DSNRGCOL   DSNRGCOL DSN_REGISTER_OB S 160314   G G G G G G G   G G
```

Figure 81. Example of using the search argument area on a table display panel

  When you press Enter, only the authorizations granted to PUBLIC are displayed:



```
ADB2AT in                  DD1A Table Authorizations           Row 1 from 1000

Commands: REVOKE  GRANT  RMIMPL
Line commands:
 R - Revoke  GR - Grant  T - Table  I - Interpretation   U   D   I S U   R   U
 CA - Column authorizations  RE - Grantee role           P A E I N E P R E   N
 ? - Show all line commands                              D L L N S L D E F T L
                                                         C T E D E E A F C R O
                     G                          H Date   O E T E R C T E O I A
S  Grantor  Grantee  T Schema   Name            G Grant  L R E X T T E R L G D
   *        PUBLIC*  * *        *               * *      * * * * * * * * * * *
-- -------- -------- - -------- --------------- - ------ - - - - - - - - - - -
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYE       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYA       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYU       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMY1       S 030921           Y
   TS5513   PUBLIC     TS5513   ADBCHG            200820           Y
   TS5513   PUBLIC     TS5513   ADBCHGS           200820           Y
   TS5513   PUBLIC     TS5513   ADBCHGSR          200820           Y
   TS5513   PUBLIC     TS5513   ADBCPREREQ        200820           Y
   TS5513   PUBLIC     TS5513   ADBCMASK          200820           Y
```

Figure 82. Filtered table display panel

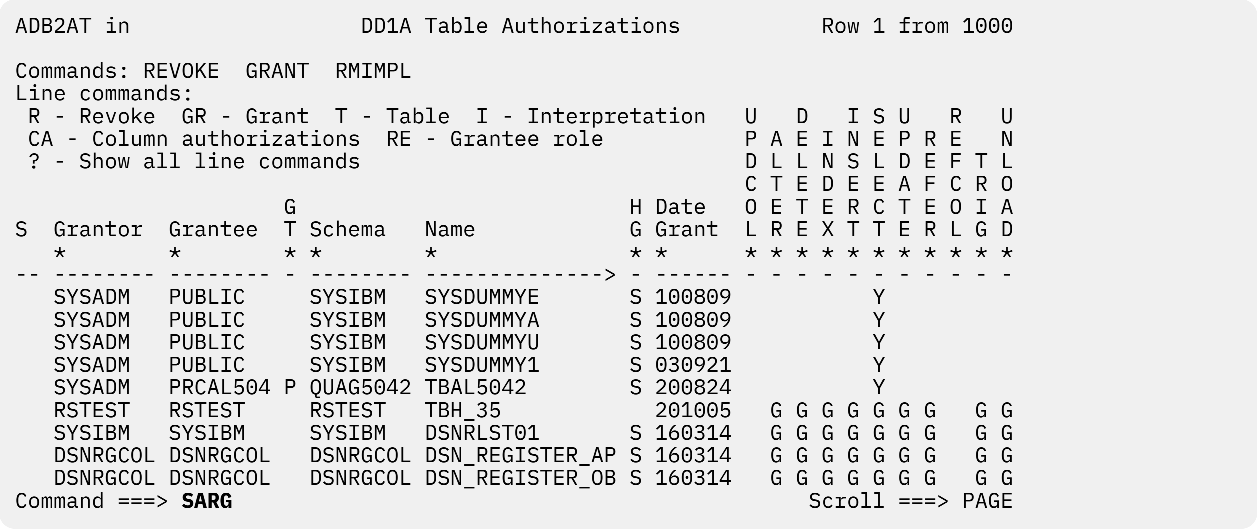
  To clear the filter, specify a blank in the search argument area, and press Enter.

  For detailed information about the search argument area and valid criteria that you can specify, see “D Search arguments” on page 193.

  Tip: Use the CAPS primary command to toggle between using mixed-case and upper-case searching. This command temporarily overrides the **Capitalize object names** setting on the **Admin Defaults (ADB2P2)** panel. For more information about CAPS, see CAPS.

- Use the SEARCH (or SARG) command to perform advanced filtering.

  For example, if you want to filter the **GT** (Grantee type) column on this same panel to see only blank values, you can specify the SARG command, and press Enter:

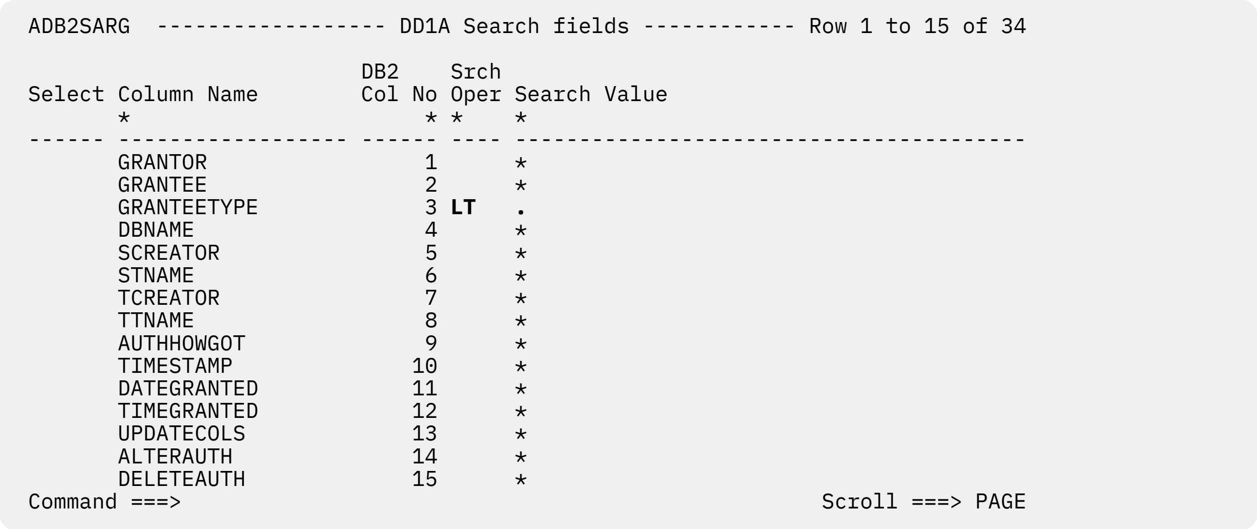


```
ADB2AT in                  DD1A Table Authorizations          Row 1 from 1000

Commands: REVOKE  GRANT  RMIMPL
Line commands:
 R - Revoke  GR - Grant  T - Table  I - Interpretation   U   D   I S U   R   U
 CA - Column authorizations  RE - Grantee role           P A E I N E P R E   N
 ? - Show all line commands                              D L L N S L D E F T L
                                                         C T E D E E A F C R O
                     G                          H Date   O E T E R C T E O I A
S  Grantor  Grantee  T Schema   Name            G Grant  L R E X T T E R L G D
   *        *        * *        *               * *      * * * * * * * * * * *
-- -------- -------- - -------- --------------> - ------ - - - - - - - - - - -
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYE       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYA       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYU       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMY1       S 030921           Y
   SYSADM   PRCAL504 P QUAG5042 TBAL5042        S 200824           Y
   RSTEST   RSTEST     RSTEST   TBH_35            201005   G G G G G G G   G G
   SYSIBM   SYSIBM     SYSIBM   DSNRLST01       S 160314   G G G G G G G   G G
   DSNRGCOL DSNRGCOL   DSNRGCOL DSN_REGISTER_AP S 160314   G G G G G G G   G G
   DSNRGCOL DSNRGCOL   DSNRGCOL DSN_REGISTER_OB S 160314   G G G G G G G   G G
Command ===> SARG                                             Scroll ===> PAGE
```

  On the subsequent **Search fields (ADB2SARG)** panel, specify the following values for GRANTEETYPE:

  – In the **SrchOper** column, specify LT

  – In the **Search Value** column, specify `.` (a period)



```
 ADB2SARG  ------------------ DD1A Search fields ------------ Row 1 to 15 of 34

                           DB2    Srch
 Select Column Name        Col No Oper Search Value
        *                       * *    *
 ------ ----------------- ------ ---- ----------------------------------------
        GRANTOR                 1      *
        GRANTEE                 2      *
        GRANTEETYPE             3 LT   .
        DBNAME                  4      *
        SCREATOR                5      *
        STNAME                  6      *
        TCREATOR                7      *
        TTNAME                  8      *
        AUTHHOWGOT              9      *
        TIMESTAMP              10      *
        DATEGRANTED            11      *
        TIMEGRANTED            12      *
        UPDATECOLS             13      *
        ALTERAUTH              14      *
        DELETEAUTH             15      *
 Command ===>                                                  Scroll ===> PAGE
```

Figure 83. ***Search fields (ADB2SARG)*** *panel*

  The following search operators are valid on this panel:

  EQ or =
  Equal to

  GT or >
  Greater than

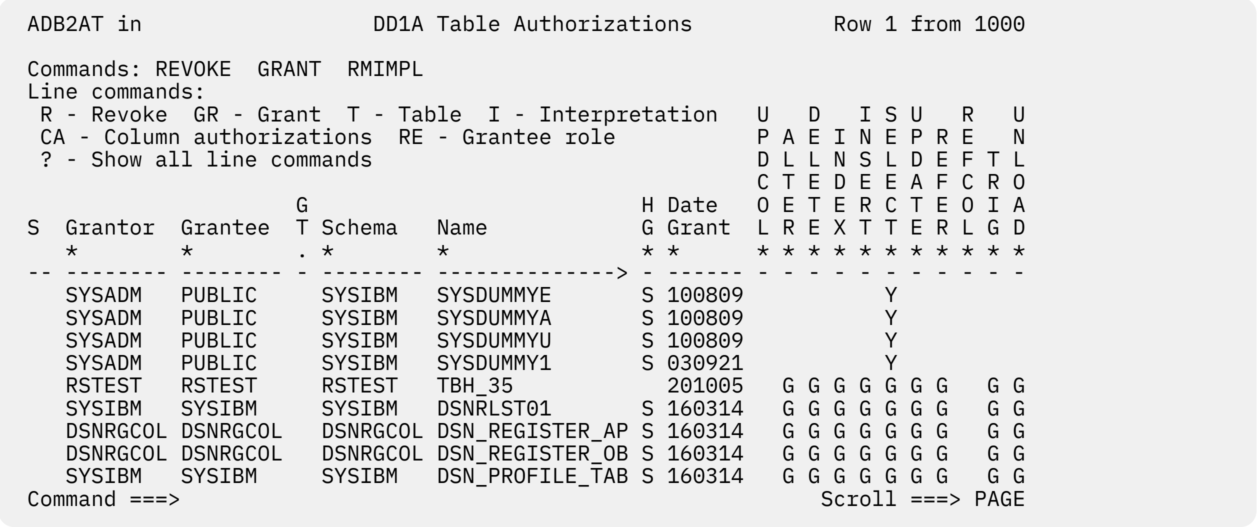
  GE or >=
  Greater than or equal to

  LT or <
  Less than

  LE or <=
  Less than or equal to

NE or ¬=
Not equal to

Press Enter. Then press END (F3) to exit the **Search fields (ADB2SARG)** panel. The previous panel displays only those rows where the **GT** column is blank:



```
ADB2AT in                  DD1A Table Authorizations           Row 1 from 1000

Commands: REVOKE  GRANT  RMIMPL
Line commands:
 R - Revoke  GR - Grant  T - Table  I - Interpretation   U   D   I S U   R   U
 CA - Column authorizations  RE - Grantee role           P A E I N E P R E   N
 ? - Show all line commands                              D L L N S L D E F T L
                                                         C T E D E E A F C R O
                     G                          H Date   O E T E R C T E O I A
S  Grantor  Grantee  T Schema   Name            G Grant  L R E X T T E R L G D
   *        *        . *        *               * *      * * * * * * * * * * *
-- -------- -------- - -------- --------------> - ------ - - - - - - - - - - -
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYE       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYA       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMYU       S 100809           Y
   SYSADM   PUBLIC     SYSIBM   SYSDUMMY1       S 030921           Y
   RSTEST   RSTEST     RSTEST   TBH_35            201005   G G G G G G G   G G
   SYSIBM   SYSIBM     SYSIBM   DSNRLST01       S 160314   G G G G G G G   G G
   DSNRGCOL DSNRGCOL   DSNRGCOL DSN_REGISTER_AP S 160314   G G G G G G G   G G
   DSNRGCOL DSNRGCOL   DSNRGCOL DSN_REGISTER_OB S 160314   G G G G G G G   G G
   SYSIBM   SYSIBM     SYSIBM   DSN_PROFILE_TAB S 160314   G G G G G G G   G G
Command ===>                                                  Scroll ===> PAGE
```

Sorting display data

On table display panels, you can sort the rows in ascending or descending order based on one or more columns.

Procedure

To sort display data, specify the SORT primary command. You can use any of the following methods to specify this command:

- **To sort on only one column:** specify the SORT primary command, put your cursor in the column to be sorted, and press Enter.

  By default, the sorting order is ascending.
- **To sort on one or more columns:** specify the SORT primary command with one or more column names (or column header names).

  For example, you can specify the following SORT command to sort the table on the **Grant date** column:

  ```
  SORT GRANTDATE
  ```

  And you can specify the following SORT command to sort the table first on the **Grant date** column and then on the **Grantee** column:

  ```
  SORT GRANTDATE GRANTEE
  ```
- **To sort on one or more columns or specify a different sorting order (ascending or descending):** specify the SORT command with no parameters, and press Enter. On the **Sort fields (ADB2SORT)** panel, specify your sorting criteria.

  In the **Select** column, specify numbers next to the appropriate columns to indicate the sorting sequence. The specified number must be greater than 0 (zero). You can also specify the A and D line commands to indicate whether you want the column sorted in ascending (A) or descending (D) order.

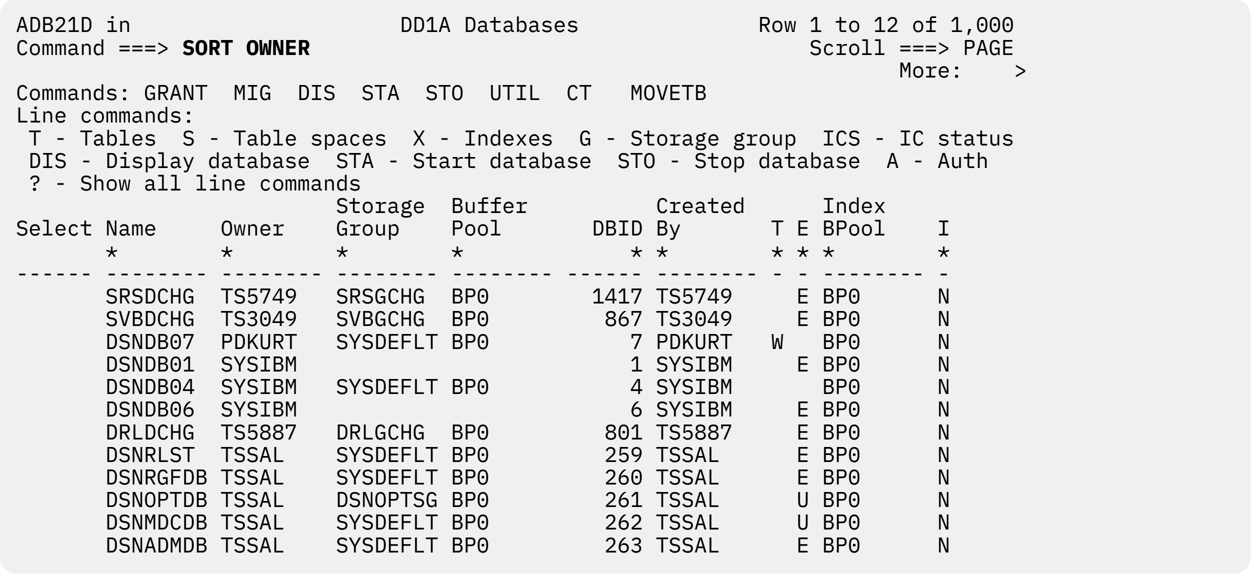
  When you exit the **Sort fields (ADB2SORT)** panel, the data is sorted according to your specifications.
- **Tip:** To save your sort specifications for a panel so that the sequence is displayed every time that the panel is displayed (until specifically deleted), issue the SORT command without any parameters. Then,

on the **Sort fields (ADB2SORT)** panel, issue the SAVE command. The saved sort sequence for each panel is saved in an ISPF table named ADBSORT in the user ISPPROF data set.

Examples

Example of sorting on one column

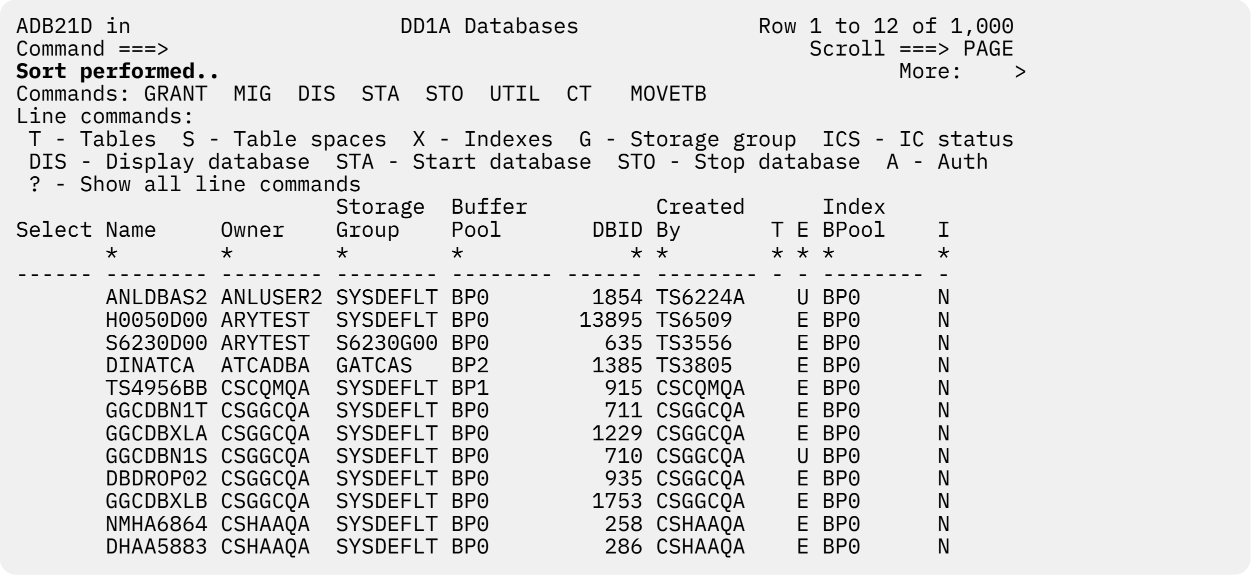
Suppose you want to sort rows on the **Databases (ADB21D)** panel based on the **Owner** column. Specify the following primary command:



```
ADB21D in                     DD1A Databases              Row 1 to 12 of 1,000
Command ===> SORT OWNER                                       Scroll ===> PAGE
                                                                     More:    >
Commands: GRANT  MIG  DIS  STA  STO  UTIL  CT   MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       *        *        *        *             * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       SRSDCHG  TS5749   SRSGCHG  BP0        1417 TS5749     E BP0      N
       SVBDCHG  TS3049   SVBGCHG  BP0         867 TS3049     E BP0      N
       DSNDB07  PDKURT   SYSDEFLT BP0           7 PDKURT   W   BP0      N
       DSNDB01  SYSIBM                          1 SYSIBM     E BP0      N
       DSNDB04  SYSIBM   SYSDEFLT BP0           4 SYSIBM       BP0      N
       DSNDB06  SYSIBM                          6 SYSIBM     E BP0      N
       DRLDCHG  TS5887   DRLGCHG  BP0         801 TS5887     E BP0      N
       DSNRLST  TSSAL    SYSDEFLT BP0         259 TSSAL      E BP0      N
       DSNRGFDB TSSAL    SYSDEFLT BP0         260 TSSAL      E BP0      N
       DSNOPTDB TSSAL    DSNOPTSG BP0         261 TSSAL      U BP0      N
       DSNMDCDB TSSAL    SYSDEFLT BP0         262 TSSAL      U BP0      N
       DSNADMDB TSSAL    SYSDEFLT BP0         263 TSSAL      E BP0      N
```

*Figure 84. Sorting the **Databases (ADB21D)** panel*

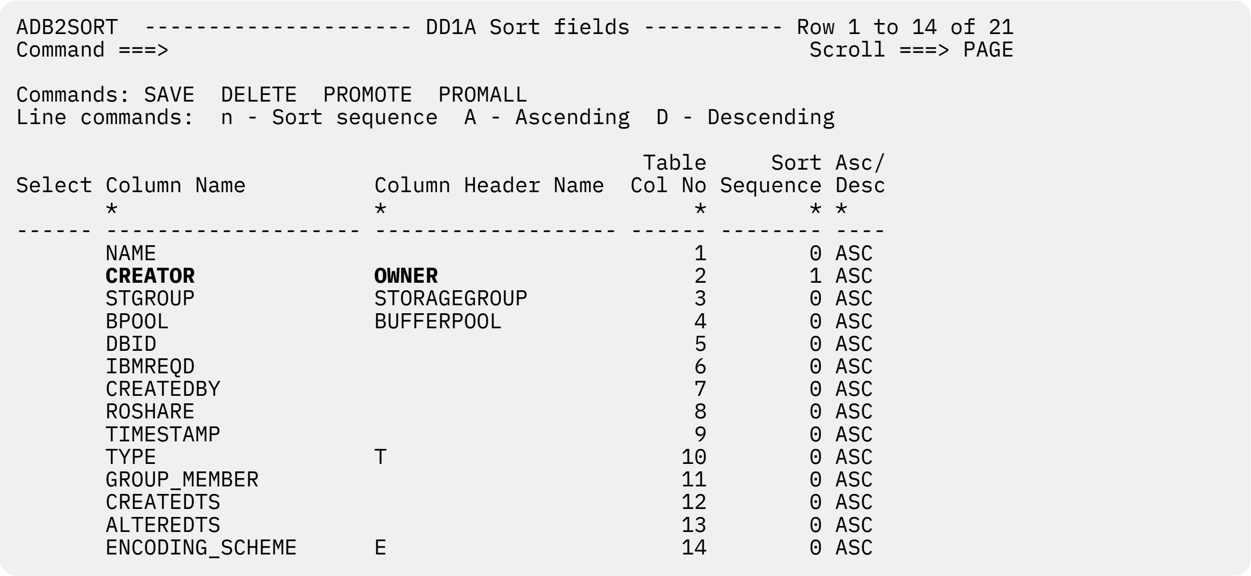
After you press Enter, you can see the sorted data:



```
ADB21D in                     DD1A Databases              Row 1 to 12 of 1,000
Command ===>                                                  Scroll ===> PAGE
Sort performed..                                                     More:    >
Commands: GRANT  MIG  DIS  STA  STO  UTIL  CT   MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       *        *        *        *             * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       ANLDBAS2 ANLUSER2 SYSDEFLT BP0        1854 TS6224A    U BP0      N
       H0050D00 ARYTEST  SYSDEFLT BP0       13895 TS6509     E BP0      N
       S6230D00 ARYTEST  S6230G00 BP0         635 TS3556     E BP0      N
       DINATCA  ATCADBA  GATCAS   BP2        1385 TS3805     E BP0      N
       TS4956BB CSCQMQA  SYSDEFLT BP1         915 CSCQMQA    E BP0      N
       GGCDBN1T CSGGCQA  SYSDEFLT BP0         711 CSGGCQA    E BP0      N
       GGCDBXLA CSGGCQA  SYSDEFLT BP0        1229 CSGGCQA    E BP0      N
       GGCDBN1S CSGGCQA  SYSDEFLT BP0         710 CSGGCQA    U BP0      N
       DBDROP02 CSGGCQA  SYSDEFLT BP0         935 CSGGCQA    E BP0      N
       GGCDBXLB CSGGCQA  SYSDEFLT BP0        1753 CSGGCQA    E BP0      N
       NMHA6864 CSHAAQA  SYSDEFLT BP0         258 CSHAAQA    E BP0      N
       DHAA5883 CSHAAQA  SYSDEFLT BP0         286 CSHAAQA    E BP0      N
```

*Figure 85. **Databases (ADB21D)** panel after SORT OWNER issued*

Note: You could also specify `SORT  CREATOR` and get the same result, because CREATOR is the column name, and OWNER is the column header name. You can specify either one. To see column names and column header names, issue the SORT command without any parameters:

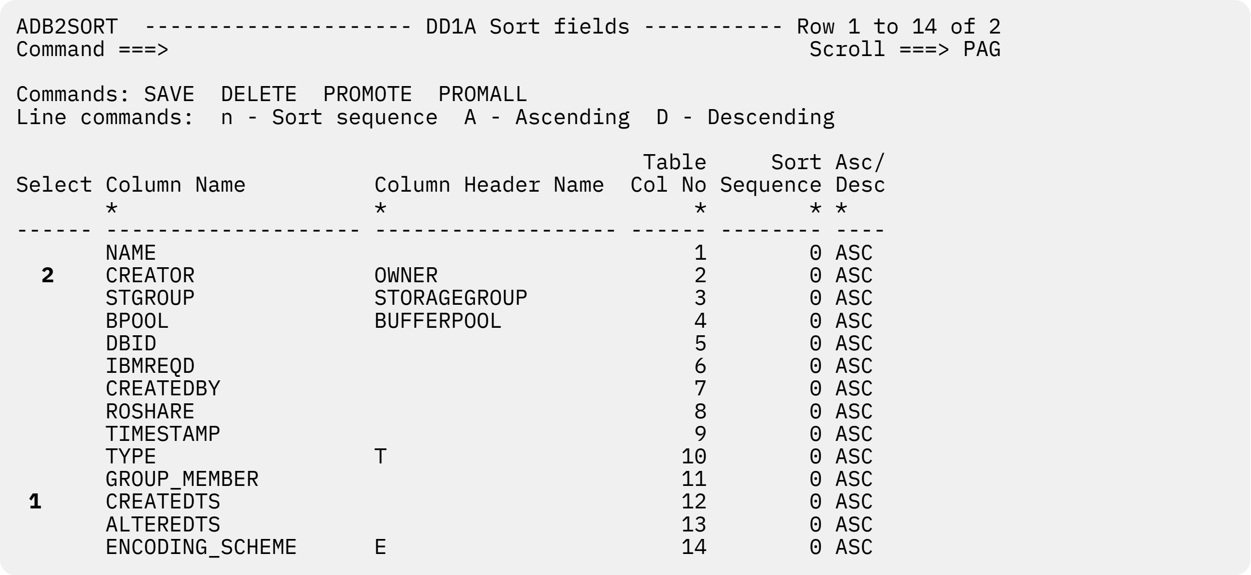


```
ADB2SORT  --------------------- DD1A Sort fields ----------- Row 1 to 14 of 21
Command ===>                                                  Scroll ===> PAGE

Commands: SAVE  DELETE  PROMOTE  PROMALL
Line commands:  n - Sort sequence  A - Ascending  D - Descending

                                                 Table     Sort Asc/
Select Column Name          Column Header Name  Col No Sequence Desc
       *                    *                        *        * *
------ -------------------- ------------------- ------ -------- ----
       NAME                                          1        0 ASC
       CREATOR              OWNER                    2        1 ASC
       STGROUP              STORAGEGROUP             3        0 ASC
       BPOOL                BUFFERPOOL               4        0 ASC
       DBID                                          5        0 ASC
       IBMREQD                                       6        0 ASC
       CREATEDBY                                     7        0 ASC
       ROSHARE                                       8        0 ASC
       TIMESTAMP                                     9        0 ASC
       TYPE                 T                       10        0 ASC
       GROUP_MEMBER                                 11        0 ASC
       CREATEDTS                                    12        0 ASC
       ALTEREDTS                                    13        0 ASC
       ENCODING_SCHEME      E                       14        0 ASC
```

Figure 86. ***Sort fields (ADB2SORT)*** *panel*

Example of sorting on multiple columns

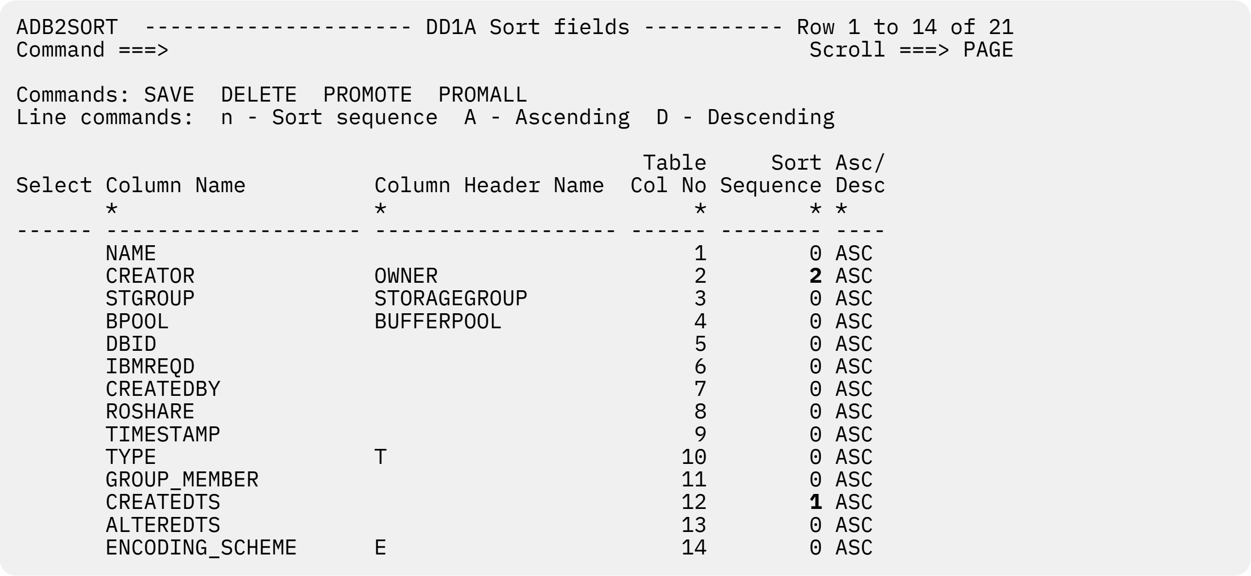
Suppose you want to sort rows on the **Databases (ADB21D)** panel based on several columns. To do so, specify the SORT command. Then, on the **Sort fields (ADB2SORT)** panel, specify your criteria. For example, the following specification indicates that the panel is to be sorted first by the CREATEDTS column and secondly by the CREATOR column:



```
ADB2SORT  --------------------- DD1A Sort fields ----------- Row 1 to 14 of 2
Command ===>                                                  Scroll ===> PAG

Commands: SAVE  DELETE  PROMOTE  PROMALL
Line commands:  n - Sort sequence  A - Ascending  D - Descending

                                                 Table     Sort Asc/
Select Column Name          Column Header Name  Col No Sequence Desc
       *                    *                        *        * *
------ -------------------- ------------------- ------ -------- ----
       NAME                                          1        0 ASC
  2    CREATOR              OWNER                    2        0 ASC
       STGROUP              STORAGEGROUP             3        0 ASC
       BPOOL                BUFFERPOOL               4        0 ASC
       DBID                                          5        0 ASC
       IBMREQD                                       6        0 ASC
       CREATEDBY                                     7        0 ASC
       ROSHARE                                       8        0 ASC
       TIMESTAMP                                     9        0 ASC
       TYPE                 T                       10        0 ASC
       GROUP_MEMBER                                 11        0 ASC
 1     CREATEDTS                                    12        0 ASC
       ALTEREDTS                                    13        0 ASC
       ENCODING_SCHEME      E                       14        0 ASC
```

Figure 87. ***Sort fields (ADB2SORT)*** *panel with sort sequence specification*

After you press Enter, you can see the updated sorting sequence:

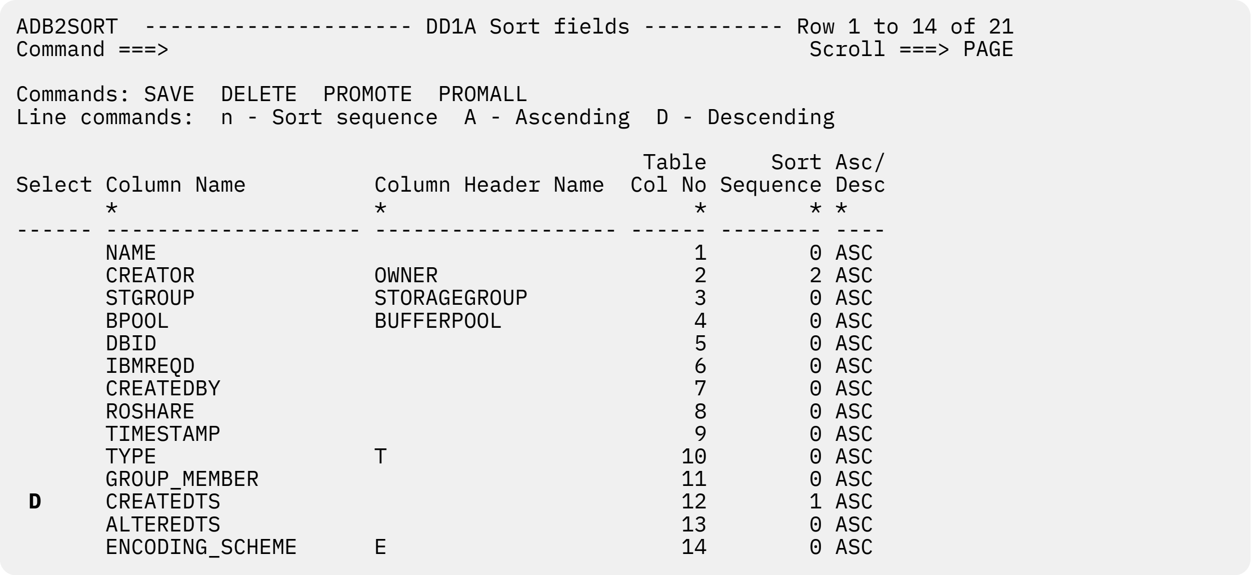


```
ADB2SORT  --------------------- DD1A Sort fields ----------- Row 1 to 14 of 21
Command ===>                                                  Scroll ===> PAGE

Commands: SAVE  DELETE  PROMOTE  PROMALL
Line commands:  n - Sort sequence  A - Ascending  D - Descending

                                                 Table     Sort Asc/
Select Column Name          Column Header Name  Col No Sequence Desc
       *                    *                        *        * *
------ -------------------- ------------------- ------ -------- ----
       NAME                                          1        0 ASC
       CREATOR              OWNER                    2        2 ASC
       STGROUP              STORAGEGROUP             3        0 ASC
       BPOOL                BUFFERPOOL               4        0 ASC
       DBID                                          5        0 ASC
       IBMREQD                                       6        0 ASC
       CREATEDBY                                     7        0 ASC
       ROSHARE                                       8        0 ASC
       TIMESTAMP                                     9        0 ASC
       TYPE                 T                       10        0 ASC
       GROUP_MEMBER                                 11        0 ASC
       CREATEDTS                                    12        1 ASC
       ALTEREDTS                                    13        0 ASC
       ENCODING_SCHEME      E                       14        0 ASC
```

Figure 88. ***Sort fields (ADB2SORT)*** *panel with updated sorting order*

A 0 in the **Sort Sequence** column indicates that the row is not part of the sorting sequence.

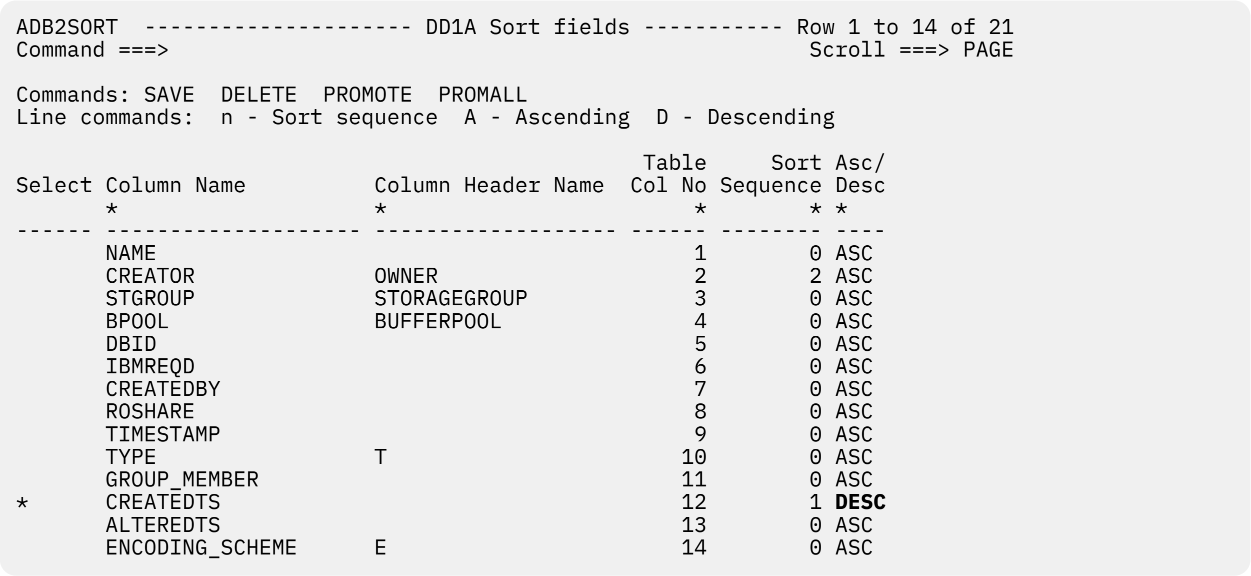
The following specification then changes the sorting order for the CREATEDTS column to descending:



```
ADB2SORT  --------------------- DD1A Sort fields ----------- Row 1 to 14 of 21
Command ===>                                                  Scroll ===> PAGE

Commands: SAVE  DELETE  PROMOTE  PROMALL
Line commands:  n - Sort sequence  A - Ascending  D - Descending

                                                 Table     Sort Asc/
Select Column Name          Column Header Name  Col No Sequence Desc
       *                    *                        *        * *
------ -------------------- ------------------- ------ -------- ----
       NAME                                          1        0 ASC
       CREATOR              OWNER                    2        2 ASC
       STGROUP              STORAGEGROUP             3        0 ASC
       BPOOL                BUFFERPOOL               4        0 ASC
       DBID                                          5        0 ASC
       IBMREQD                                       6        0 ASC
       CREATEDBY                                     7        0 ASC
       ROSHARE                                       8        0 ASC
       TIMESTAMP                                     9        0 ASC
       TYPE                 T                       10        0 ASC
       GROUP_MEMBER                                 11        0 ASC
 D     CREATEDTS                                    12        1 ASC
       ALTEREDTS                                    13        0 ASC
       ENCODING_SCHEME      E                       14        0 ASC
```

Figure 89. ***Sort fields (ADB2SORT)*** *panel with sort order specification*

After you press Enter, you can see the updated sorting order:



```
ADB2SORT  --------------------- DD1A Sort fields ----------- Row 1 to 14 of 21
Command ===>                                                  Scroll ===> PAGE

Commands: SAVE  DELETE  PROMOTE  PROMALL
Line commands:  n - Sort sequence  A - Ascending  D - Descending

                                                 Table     Sort Asc/
Select Column Name          Column Header Name  Col No Sequence Desc
       *                    *                        *        * *
------ -------------------- ------------------- ------ -------- ----
       NAME                                          1        0 ASC
       CREATOR              OWNER                    2        2 ASC
       STGROUP              STORAGEGROUP             3        0 ASC
       BPOOL                BUFFERPOOL               4        0 ASC
       DBID                                          5        0 ASC
       IBMREQD                                       6        0 ASC
       CREATEDBY                                     7        0 ASC
       ROSHARE                                       8        0 ASC
       TIMESTAMP                                     9        0 ASC
       TYPE                 T                       10        0 ASC
       GROUP_MEMBER                                 11        0 ASC
*      CREATEDTS                                    12        1 DESC
       ALTEREDTS                                    13        0 ASC
       ENCODING_SCHEME      E                       14        0 ASC
```

*Figure 90. **Sort fields (ADB2SORT)** panel with updated sorting order*

Refreshing data on panels

As you work through Db2 Admin Tool panels, you might want to refresh the original data on a panel after specifying new data or changing data on that panel.

Procedure

To refresh the data, specify the REFRESH primary command, and press Enter.

Displaying and entering long field values

Some fields do not display complete values because the space on the panel is limited. In these cases, you can use the ISPF scrollable fields to see more content.

About this task

Scrollable fields are denoted by the less than (<) and greater than (>) symbols. A > symbol indicates that the field can be scrolled to the right. A < symbol indicates that the field can be scrolled to the left. Both symbols are displayed when the field can scroll either left or right.

Procedure

To display and enter long field values, interact with these scrollable fields in the following ways:

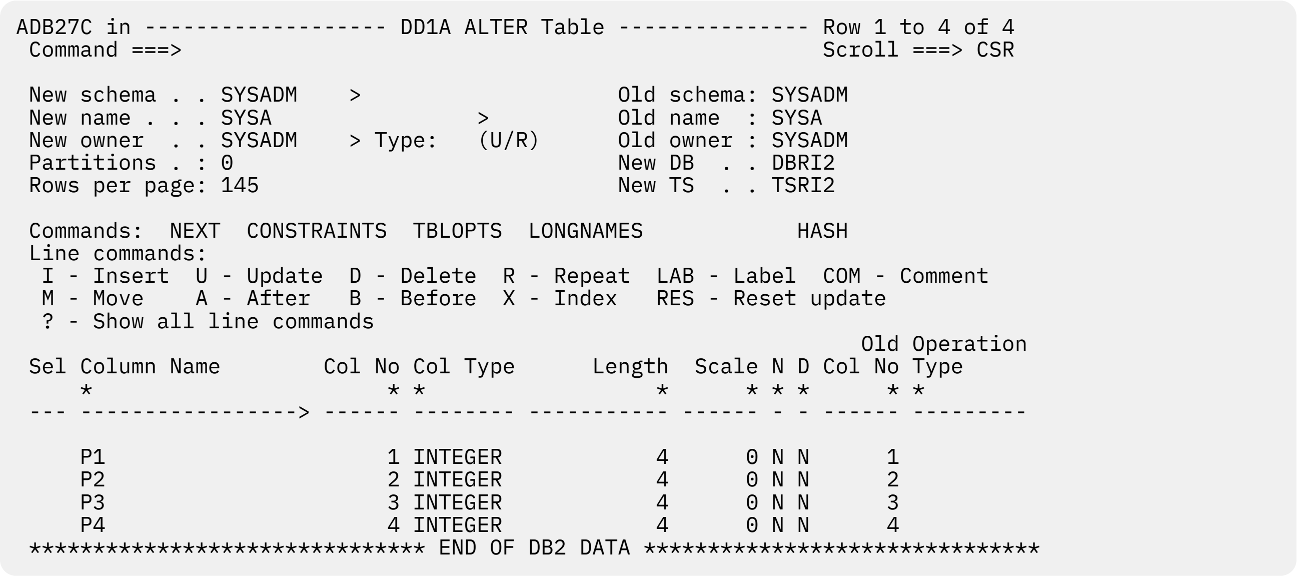
- To scroll through the field, type LEFT or RIGHT in the command field, position the cursor in the field, and press Enter.
- To see the entire contents of the field at once or enter a value that is longer than the field, type EXPAND in the command field, position your cursor in the scrollable field, and press Enter.
- To clear the contents of the field, type ZCLRSFLD in the command field, position your cursor in the scrollable field, and press Enter. (If your level of z/OS does not support the ZCLRSFLD command, you can use the EXPAND command to display the entire contents of the field, and then clear the contents of the field in the pop-up window.)

  Tip: You can assign your PF keys to be the LEFT, RIGHT, EXPAND, and ZCLRSFLD commands. Using a PF key simulates both typing in the command and pressing Enter.

Example

Example of scrolling on the ALTER Table (ADB27C) panel:

On the **ALTER Table (ADB27C)** panel, **New schema** and **New name** are scrollable input fields. **Old schema** and **Old name** are scrollable output fields if the value is long enough to require scrolling. **Column Name** is a scrollable input and output column. (To scroll the **Column Name** column, you need to position your cursor on a column name value, such as P1 in the following screen.)

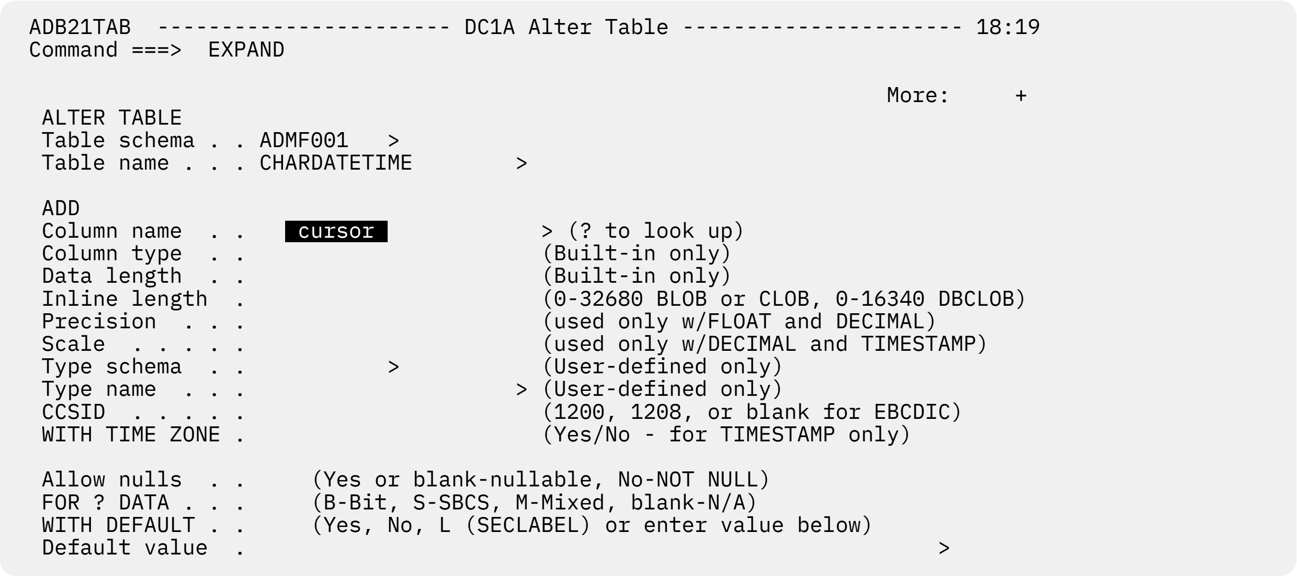


```
ADB27C in ------------------- DD1A ALTER Table --------------- Row 1 to 4 of 4
 Command ===>                                                  Scroll ===> CSR

 New schema . . SYSADM    >                   Old schema: SYSADM
 New name . . . SYSA                >         Old name  : SYSA
 New owner  . . SYSADM    > Type:   (U/R)     Old owner : SYSADM
 Partitions . : 0                             New DB   . . DBRI2
 Rows per page: 145                           New TS   . . TSRI2

 Commands:  NEXT  CONSTRAINTS  TBLOPTS  LONGNAMES            HASH
 Line commands:
  I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
  M - Move    A - After   B - Before  X - Index   RES - Reset update
  ? - Show all line commands
                                                                  Old Operation
 Sel Column Name        Col No Col Type      Length  Scale N D Col No Type
     *                       * *                  *      * * *      * *
 --- ----------------> ------ -------- ---------- ------ - - ------ ---------

     P1                     1 INTEGER            4      0 N N      1
     P2                     2 INTEGER            4      0 N N      2
     P3                     3 INTEGER            4      0 N N      3
     P4                     4 INTEGER            4      0 N N      4
 ******************************* END OF DB2 DATA *******************************
```

*Figure 91. Scrollable fields on the **ALTER Table (ADB27C)** panel*

Example of entering a long column name on the Alter Table (ADB21TAB) panel:

On the **Alter Table (ADB21TAB)** panel, you can enter a long column name by typing EXPAND on the command line, placing the cursor in the **Column name** field, and pressing Enter. You can then enter a long name on panel ISPEXPAND.



```
 ADB21TAB  ----------------------- DC1A Alter Table ---------------------- 18:19
 Command ===>  EXPAND

                                                                    More:     +
  ALTER TABLE
  Table schema . . ADMF001   >
  Table name . . . CHARDATETIME        >

  ADD
  Column name  . .   cursor             > (? to look up)
  Column type  . .                        (Built-in only)
  Data length  . .                        (Built-in only)
  Inline length  .                        (0-32680 BLOB or CLOB, 0-16340 DBCLOB)
  Precision  . . .                        (used only w/FLOAT and DECIMAL)
  Scale  . . . . .                        (used only w/DECIMAL and TIMESTAMP)
  Type schema  . .           >            (User-defined only)
  Type name  . . .                      > (User-defined only)
  CCSID  . . . . .                        (1200, 1208, or blank for EBCDIC)
  WITH TIME ZONE .                        (Yes/No - for TIMESTAMP only)

  Allow nulls  . .     (Yes or blank-nullable, No-NOT NULL)
  FOR ? DATA . . .     (B-Bit, S-SBCS, M-Mixed, blank-N/A)
  WITH DEFAULT . .     (Yes, No, L (SECLABEL) or enter value below)
  Default value  .                                                    >
```

*Figure 92. Scrollable fields on the **Alter Table (ADB21TAB)** panel*

Setting panel display options

You can customize settings across all of the Db2 Admin Tool panels that display lists of objects.

Procedure

1. Issue the OPTIONS D command.

   The **Panel Display Options** pop-up window is displayed.
2. Select which of the following items you want to be included on the panel by specifying a slash (/) in front of the item, and press the Enter key.

   - Db2 Admin Tool action bar. When the action bar is selected, the DB2 Admin Tool action bar will be shown on select panels. Not all panels support this feature.
   - Panel instructions.
   - Primary commands.
   - Line commands.
   - Filter line.

   Deselecting some of the items will result in a simpler-looking panel that displays more data objects.

Commands in Db2 Admin Tool

You can use primary commands and line commands in Db2 Admin Tool.

Primary commands

Primary commands are issued from the command line on Db2 Admin Tool panels. Some primary commands can be entered on all panels and some primary commands are restricted to certain panels. For more information about a particular primary command, see "Db2 Admin Tool primary commands" on page 211 or the Help panels.

Tips:

- When you enter a primary command that has the same name as a TSO command, the TSO command is executed first. To bypass the TSO command processor, enter the primary command with a prefix of the greater than symbol (>), which is a TSO escape character.
- On table display panels, primary commands operate on each row that is displayed in the table. If you want to omit some of the rows before you issue the primary command, use the minus (-) line command to remove rows from the display. The primary commands operate only on rows that are listed.

Line commands

Line commands specify an operation that is to be performed on the selected line or object. You can issue line commands from ISPF table display panels. Specify line commands in the line command area in front of each row. This area is called the SELECT (or SEL) field.

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

Two types of line commands are supported: special line commands and general line commands. You also can define your own line commands during the installation process.

Special line commands

The special line commands that are available for a panel are listed in the line command description area.

A question mark (?) line command indicates that not all line commands are listed because of limited space. Enter ? in the **Select** column to display a list of all of the line commands available for that panel.

Because the objects that are listed on a panel have varying attributes, not all of the line commands that are shown on the panel or its extension panel are applicable to each object. When you try to issue a line command against an object where it is not applicable, an error message is returned.

Utility line commands, which open Db2 utility panels, are prefixed with "U."

For information about a particular line command on a panel, see the online help (PF1).

General line commands

The following general line commands are valid on any table display panel:

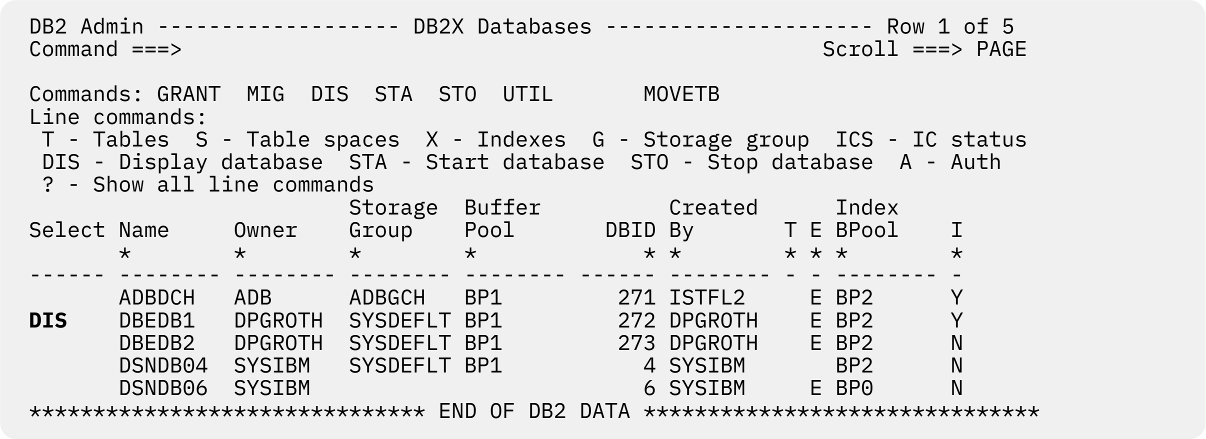
Minus (-)

Excludes a line from the list. You can specify multiple minus (-) line commands at a time.

Equals (=)

Repeats the last entered line command.

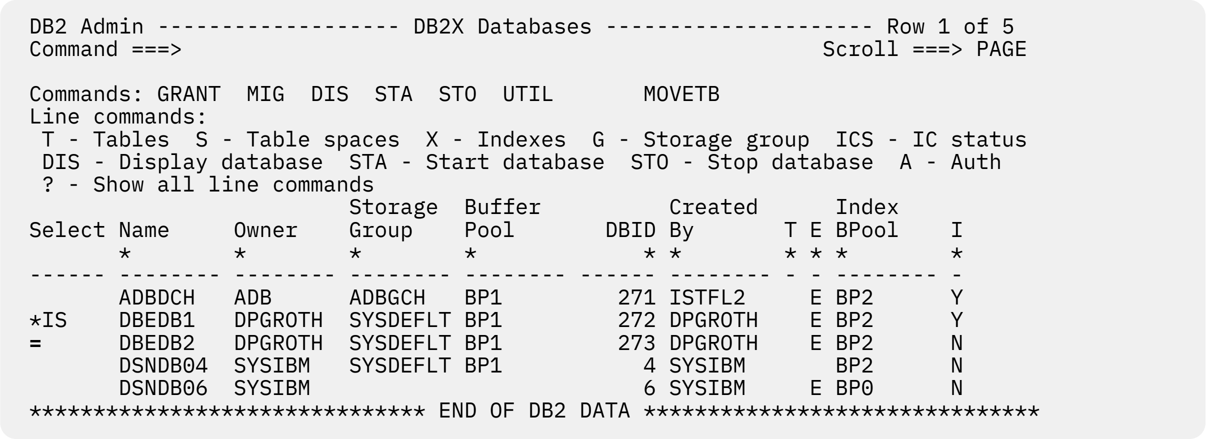
For example, in the following figure, the DIS line command is entered to request a display of the DBEDB1 database:



```
DB2 Admin ------------------- DB2X Databases --------------------- Row 1 of 5
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                        Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       *        *        *        *             * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       ADBDCH   ADB      ADBGCH   BP1         271 ISTFL2     E BP2      Y
DIS    DBEDB1   DPGROTH  SYSDEFLT BP1         272 DPGROTH    E BP2      Y
       DBEDB2   DPGROTH  SYSDEFLT BP1         273 DPGROTH    E BP2      N
       DSNDB04  SYSIBM   SYSDEFLT BP1           4 SYSIBM       BP2      N
       DSNDB06  SYSIBM                          6 SYSIBM     E BP0      N
******************************* END OF DB2 DATA *******************************
```

*Figure 93. DIS line command on the **Databases (ADB21D)** panel*

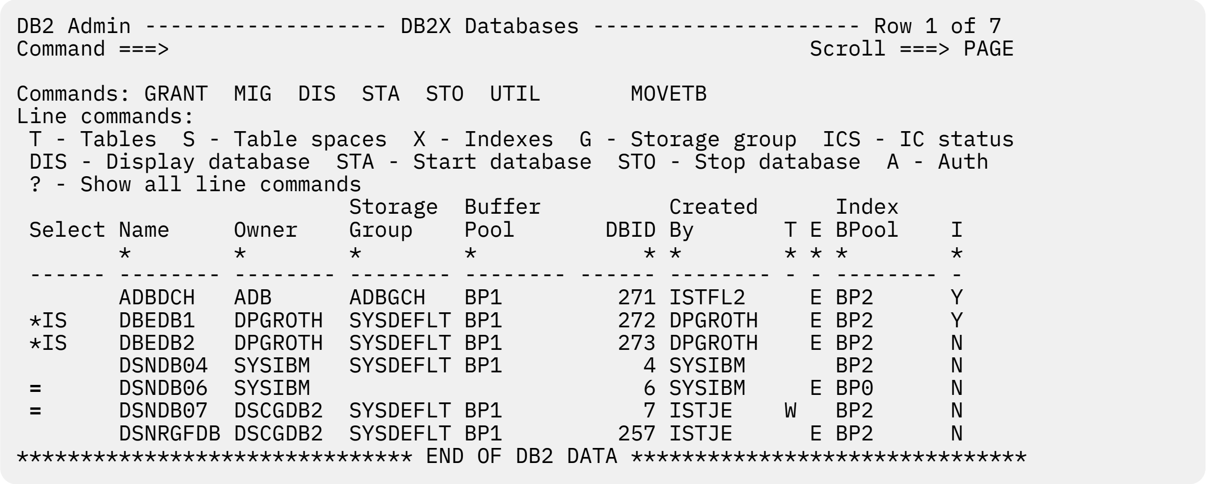
After Db2 Admin Tool executes this line command, the first character of the command is replaced with an asterisk (*). If you then specify = for the next line and press Enter, the DIS line command is executed for DBEDB2 database:



```
DB2 Admin ------------------- DB2X Databases --------------------- Row 1 of 5
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                        Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       *        *        *        *             * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       ADBDCH   ADB      ADBGCH   BP1         271 ISTFL2     E BP2      Y
*IS    DBEDB1   DPGROTH  SYSDEFLT BP1         272 DPGROTH    E BP2      Y
=      DBEDB2   DPGROTH  SYSDEFLT BP1         273 DPGROTH    E BP2      N
       DSNDB04  SYSIBM   SYSDEFLT BP1           4 SYSIBM       BP2      N
       DSNDB06  SYSIBM                          6 SYSIBM     E BP0      N
******************************* END OF DB2 DATA *******************************
```

*Figure 94. Equals (=) line command on the **Databases (ADB21D)** panel*

If you enter the = line command multiple times, Db2 Admin Tool issues each line command successively. The panel where you entered the = line commands is not shown between executions of the line commands. In the following example, Db2 Admin Tool displays the DSNDB06 database first. When you exit that display, Db2 Admin Tool immediately displays the DSNDB07 database.



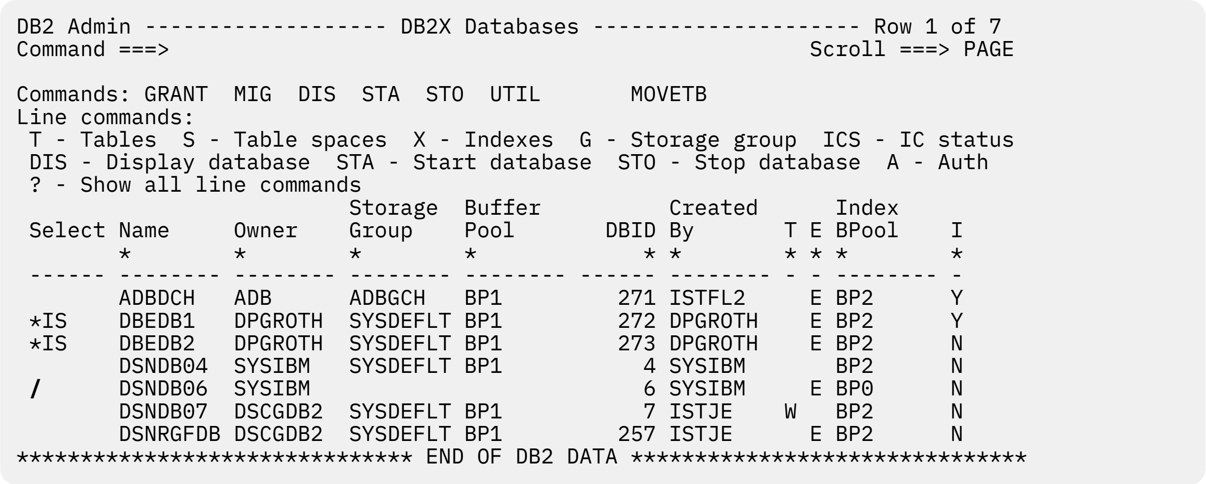
```
DB2 Admin ------------------- DB2X Databases --------------------- Row 1 of 7
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                          Storage  Buffer          Created      Index
 Select Name     Owner    Group    Pool       DBID By       T E BPool    I
        *        *        *        *             * *        * * *        *
 ------ -------- -------- -------- -------- ------ -------- - - -------- -
        ADBDCH   ADB      ADBGCH   BP1         271 ISTFL2     E BP2      Y
 *IS    DBEDB1   DPGROTH  SYSDEFLT BP1         272 DPGROTH    E BP2      Y
 *IS    DBEDB2   DPGROTH  SYSDEFLT BP1         273 DPGROTH    E BP2      N
        DSNDB04  SYSIBM   SYSDEFLT BP1           4 SYSIBM       BP2      N
 =      DSNDB06  SYSIBM                          6 SYSIBM     E BP0      N
 =      DSNDB07  DSCGDB2  SYSDEFLT BP1           7 ISTJE    W   BP2      N
        DSNRGFDB DSCGDB2  SYSDEFLT BP1         257 ISTJE      E BP2      N
******************************* END OF DB2 DATA *******************************
```

Figure 95. Multiple equals (=) line commands ***Databases (ADB21D)*** *panel*

Slash (/)

Shows all column names and their values for the selected row. You can enter more than one slash (/) line command at a time, with the exception of the **Launchpad (ADBDMT)** panel. On that panel, you can specify only one / line command at a time.

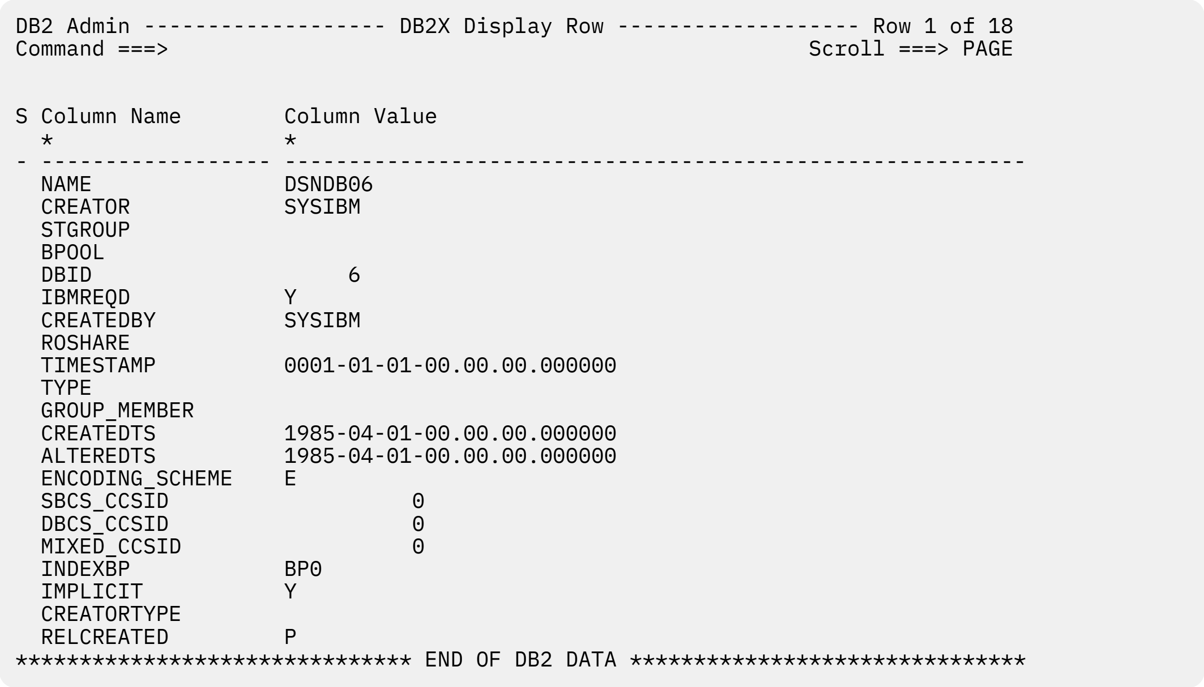
For example, on the following **Databases (ADB21D)** panel, suppose that you issue the / line command for database DSNDB06:



```
DB2 Admin ------------------- DB2X Databases --------------------- Row 1 of 7
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                          Storage  Buffer          Created      Index
 Select Name     Owner    Group    Pool       DBID By       T E BPool    I
        *        *        *        *             * *        * * *        *
 ------ -------- -------- -------- -------- ------ -------- - - -------- -
        ADBDCH   ADB      ADBGCH   BP1         271 ISTFL2     E BP2      Y
 *IS    DBEDB1   DPGROTH  SYSDEFLT BP1         272 DPGROTH    E BP2      Y
 *IS    DBEDB2   DPGROTH  SYSDEFLT BP1         273 DPGROTH    E BP2      N
        DSNDB04  SYSIBM   SYSDEFLT BP1           4 SYSIBM       BP2      N
 /      DSNDB06  SYSIBM                          6 SYSIBM     E BP0      N
        DSNDB07  DSCGDB2  SYSDEFLT BP1           7 ISTJE    W   BP2      N
        DSNRGFDB DSCGDB2  SYSDEFLT BP1         257 ISTJE      E BP2      N
******************************* END OF DB2 DATA *******************************
```

Figure 96. Slash (/) line command on the ***Databases (ADB21D)*** *panel*

Db2 Admin Tool displays all column names and their values from the SYSIBM.SYSDATABASE catalog table:



```
DB2 Admin ------------------- DB2X Display Row ------------------- Row 1 of 18
Command ===>                                                  Scroll ===> PAGE


S Column Name        Column Value
  *                  *
- ------------------ ----------------------------------------------------------
  NAME               DSNDB06
  CREATOR            SYSIBM
  STGROUP
  BPOOL
  DBID                    6
  IBMREQD            Y
  CREATEDBY          SYSIBM
  ROSHARE
  TIMESTAMP          0001-01-01-00.00.00.000000
  TYPE
  GROUP_MEMBER
  CREATEDTS          1985-04-01-00.00.00.000000
  ALTEREDTS          1985-04-01-00.00.00.000000
  ENCODING_SCHEME    E
  SBCS_CCSID                   0
  DBCS_CCSID                   0
  MIXED_CCSID                  0
  INDEXBP            BP0
  IMPLICIT           Y
  CREATORTYPE
  RELCREATED         P
******************************* END OF DB2 DATA *******************************
```

Figure 97. Result of the slash (/) line command

Db2 Admin Tool primary commands

Primary commands are issued from the command line on Db2 Admin Tool panels.

Each primary command can be categorized as one of the following types:

- “Global primary commands” on page 211
- “Primary commands for table display panels” on page 221
- “Primary commands for specific panels” on page 223

Global primary commands

The following table lists the available primary commands that you can issue from any panel in Db2 Admin Tool.

Table 11. Primary commands that can be issued from any panel

Command	**Alias**	**Description**

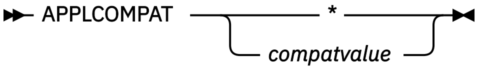
APPLCOMPAT	APPLC	Sets the current application compatibility level for dynamic SQL statements.
		This command is equivalent to issuing the SET CURRENT APPLICATION COMPATIBILITY statement.
		Syntax:
		
		compatvalue
		The application compatibility level. You can specify one of the following formats
		• V12R1M*mmm*
		• FL*mmm*
		• *mmm*
		where *mmm* is the function level number, such as 502. If you do not explicitly specify V*xx*, the current version is assumed.
		Alternatively, you can specify an asterisk (*) to set the application compatibility level to the current function level.
		The specified function level must be equal to or less than the current version and function level and the APPLCOMPAT value that was specified for ADBMAIN at bind time. The new value applies to only the current Db2 connection and does not apply to new Db2 connections that might be established during the Db2 Admin Tool session.
		Examples:
		`APPLCOMPAT V12R1M502`
		`APPLCOMPAT FL502`
		`APPLCOMPAT 502`
		`APPLCOMPAT *`
		`APPLC V12R1M502`
		`APPLC FL502`
		`APPLC 502`
		`APPLC *`

Table 11. Primary commands that can be issued from any panel (continued)

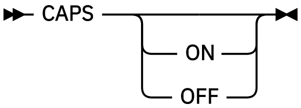
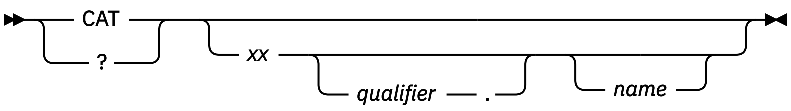
Command	Alias	Description
CAPS		Temporarily overrides the **Capitalize object names** setting on the **Admin Defaults (ADB2P2)** panel. For more information about this setting, see “Changing defaults” on page 231. **Syntax:**  **ON** Specifies that all characters are to be translated to uppercase characters. **OFF** Specifies that lowercase characters are not to be translated to uppercase characters. Specifying CAPS (without ON or OFF) toggles between CAPS ON and CAPS OFF.

Table 11. Primary commands that can be issued from any panel (continued)

Command	Alias	Description

CAT	?	Navigates directly to an object.
		A question mark (?) is the default shortcut character for the catalog navigation command, CAT. You can specify a different character on the **DB2 Admin Defaults (ADB2P2)** panel in the **CAT command character** field.
		Syntax:
		
		xx
		The object type from the **System Catalog (ADB21)** panel. For example, s is for table spaces, v is for views, and x is for indexes.
		qualifier
		The object qualifier. You can specify any value that is valid in the **Owner** field on the **System Catalog (ADB21)** panel.
		name
		The object name. You can specify any value that is valid in the **Name** field on the **System Catalog (ADB21)** panel.
		All parameters are optional. However, you must specify *xx* if you specify *qualifier* or *name*.
		Examples:
		`?d MYDB`
		`?s MYDB.TS1`
		`?s TS1`
		`?s MYDB.`
		`CAT s MYDB.TS1`
		For a demonstration of how to use this command, see Video: Catalog Navigation Command.

CATHLQ		Updates the high-level qualifier that Db2 Admin Tool is to use during the current session for data sets for the Db2 catalog. This command displays the **DB2 Space Manager (ADB2MSPC)** panel, where you can specify the qualifier.
CLREST		Restores packages that were freed by the collection clean up function. This command displays the **Restore Packages (ADBPBMRE)** panel, where you can specify options for restoring the packages. See “Restoring packages” on page 926.
CMDS		Displays the active user command list. See “Defining your own primary commands” on page 1039.
CMM		Opens the main menu for the Change Management (CM) function. This command displays the **Change Management (CM) (ADB2C)** panel, where you can choose which CM function you want to invoke.

Table 11. Primary commands that can be issued from any panel (continued)

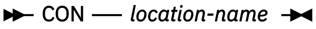
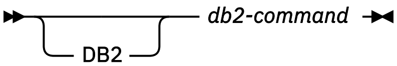
Command	**Alias**	**Description**
CON		Connects to a remote system. **Syntax:**  ***location-name*** The location name for the remote system.
DB2	*db2 command*	Issues a Db2 command. **Syntax:**  DB2 can be omitted from the command. ***db2-command*** Any valid Db2 command except for START DB2. Start the command with the command recognition character, which is usually a hyphen (-). **Example:** `DB2 -DIS THREAD (*)`
DGRP		Runs the Db2 DISPLAY GROUP command.
DTT		Displays all threads.
DUTIL		Displays a list of active and stopped utilities. This list is displayed on the **Display or Terminate Utilities (ADB2Z2U2)** panel. See "Displaying or terminating utilities" on page 902.
EXIT		Exits Db2 Admin Tool.

Table 11. Primary commands that can be issued from any panel (continued)

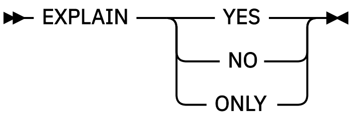
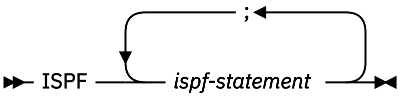
Command	Alias	Description
EXPLAIN		Changes the value of the Db2 CURRENT EXPLAIN MODE special register. Depending on your prompt options, you might be asked to confirm the change before the SET statement is executed. See "Changing Db2 Admin Tool prompt options" on page 243. **Syntax:**  **YES** Turns EXPLAIN mode on. **NO** Turns EXPLAIN mode off. **ONLY** Turns EXPLAIN mode on and prevents statements from being executed. Do not use this mode with regular Db2 Admin Tool functions. For more information about these values, see CURRENT EXPLAIN MODE special register (Db2 12 for z/OS documentation) . **Example:** `EXPLAIN YES`
ISPF		Issues one or more ISPF statements. **Syntax:**  ***ispf-statement*** A valid ISPF statement. Use a semicolon (;) to separate ISPF statements. **Example:** `ISPF SELECT CMD(MYCMD)`
LEINFO		Displays Language Environment (LE) options.
OPTIONS	OPT	Allows you to change your Db2 Admin Tool settings. This command displays the **DB2 Admin Options (ADB2P)** panel, where you can select the settings that you want to change.
OPTIONS BP	OPT BP	Allows you to change the parameters for utility batch jobs. This command displays the **Batch Job Utility Parameters (ADB2UPA)** panel, where you can specify parameters for your utility jobs.

Table 11. Primary commands that can be issued from any panel (continued)

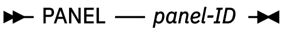
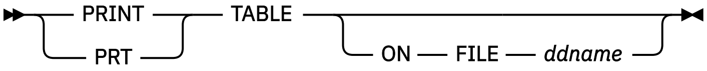
Command	Alias	Description
OPTIONS DISPLAY	OPT DISPLAY	Allows you to change the panel display. This command displays the **Panel Display Options (ADBPPDO)** panel, where you can select the items that you want displayed on panels.
PANEL		Displays the specified panel. This command allows you to extend your Db2 Admin Tool installation with your own panels and then use these panels directly with Db2 Admin Tool. You must design your panels to be invoked with this PANEL command. Your panels should not be designed to be part of a multi-panel dialog and rely on variables being set in the preceding panels. Otherwise, unpredictable results can occur. For more information about creating your own panels, see "Db2 Admin Tool application development" on page 1044. **Syntax:**  ***panel-ID*** The panel ID, such as ADB21D. **Example:** `PANEL ADBP7P`
PARMS	PARM	Displays the current Db2 Admin Tool parameters. This command displays the **Admin Defaults (ADB2P2)** panel, where you optionally make any changes to these default settings.
PRINT TABLE	PRT TABLE	Prints the current table to the specified file. **Syntax:**  ***ddname*** The DD name for the file. The specified file should be allocated exclusively, which means that it has a disposition of OLD, NEW or MOD. For example, `tso alloc f(temp1) dsn(temp1.list) old`. If you do not specify a file name, the default file with the DD name PRINT is used. **Tip:** You can use option P.P to allocate a data set for PRINT TABLE. **Example:** `PRT TABLE ON FILE temp1`

Table 11. Primary commands that can be issued from any panel (continued)

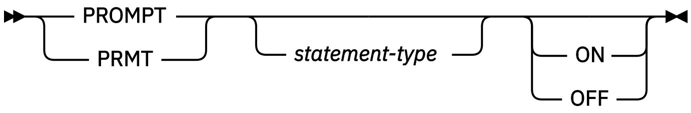
Command	Alias	Description
PROMPT (*options*)	PRMT	Changes Db2 Admin Tool prompt options. **Syntax:**  ***statement-type*** The statement type for which you want to change the prompt options. Valid values are: **DDL** *Data definition language*, or definition SQL statements. **DCL** *Data control language*, or authorization SQL statements. **DML** *Data manipulation language*, or update SQL statements. **DB2** Db2 commands. **DSN** DSN commands. **ON** Turns the prompt on for the specified statement type. If a statement type is not specified, the prompt is turned on for all statement types. **OFF** Turns the prompt off for the specified statement type. If a statement type is not specified, the prompt is turned off for all statement types. If no options are specified after PROMPT or PRMT, the **Prompt Options (ADB2PRMT)** panel is displayed, and you can enter your preferences on that panel. **Example:** `PROMPT DB2 OFF`
QTAB		Displays open ISPF tables.
REGION		Displays used and available memory (TSO region information).
SAVE		Saves the current selections or report. When SAVE is issued on the **Details for object(s) (ADBPD)** panel, the detail report (which is generated by the DET command) is saved in the specified data set.

Table 11. Primary commands that can be issued from any panel (continued)

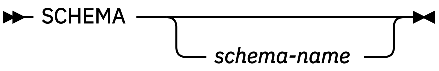
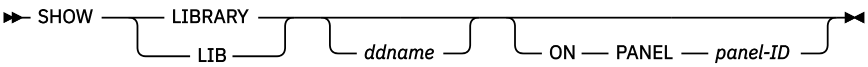
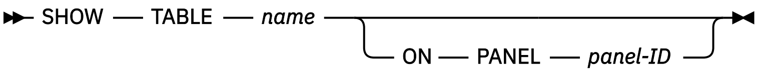
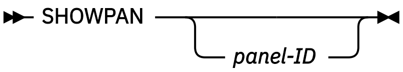
Command	Alias	Description
SCHEMA		Changes the CURRENT SCHEMA special register. Depending on your prompt options, you might be asked to confirm the change before the SET statement is executed. See "Changing Db2 Admin Tool prompt options" on page 243. **Syntax:**  ***schema-name*** The schema name. If no value is specified, CURRENT SCHEMA is set to the SQL ID. **Example:** `SCHEMA ISTJE`
SHOW LIBRARY	SHOW LIB	Shows a member list for the specified library on the specified panel. **Syntax:**  ***ddname*** The library DD name. The data set that is associated with the DD name must be a PDS or PDSE. If you do not specify a library name, the default library ISPTABL is used. **panel-ID** The panel ID. If you do not specify a panel name, the default panel DB2ADL is used.
SHOW TABLE		Shows the specified table. **Syntax:**  ***name*** The table name. The table must be in open status. **panel-ID** The panel ID. If you do not specify a panel name, the default panel ADB2DF is used.
SHOWPAN		Shows the source code for a panel. **Syntax:**  ***panel-ID*** The panel ID, such as ADB21T. The default is the current panel.

Table 11. Primary commands that can be issued from any panel (continued)

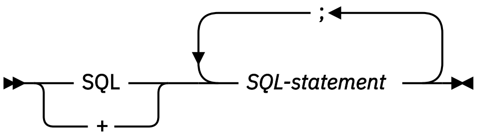
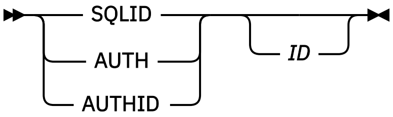
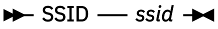
Command	**Alias**	**Description**
SMPLIST		Generates a job to run SMP/E list commands on the CSI.
SQL	+	Issues one or more SQL statements. If an SQL statement returns rows, the default table display panel shows the rows. **Syntax:**  ***SQL-statement*** An SQL statement. Use a semicolon (;) to separate SQL statements. If you do not specify an SQL statement, the **Edit/Run SQL Statement (ADBPMESQ)** panel is displayed. You can enter SQL statements on this panel and run them. **Example:** `SQL SELECT * FROM MYTABLE`
SQLID	AUTH, AUTHID	Displays or changes the current SQLID. **Syntax:**  ***ID*** The authorization ID. If you do not specify an authorization ID, the **DB2 Change Current SQL ID (ADB24)** panel is displayed. You can select a value from this panel. **Example:** `SQLID ISTJE`
SSID		Switches to another Db2 SSID. **Syntax:**  ***ssid*** The subsystem ID. **Example:** `SSID DSN9`
STATUS	STAT	Shows the current status of Db2 Admin Tool and execution control statement statistics.

Table 11. Primary commands that can be issued from any panel (continued)

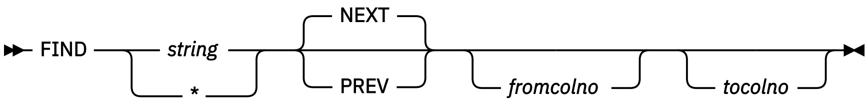
Command	Alias	Description
TESTTIME		Runs the TSO command TIME.
WSL		Opens the main menu for the Work Statement List (WSL) function. This command displays the **Manage Work Statement Lists (ADB2W)** panel, where you can select the WSL list or list library that you want to view.
ZPARM		Displays the Db2 subsystem parameter (zparm) values for the current subsystem.

Primary commands for table display panels

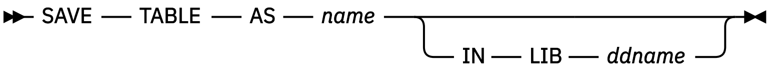
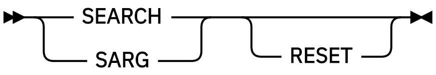
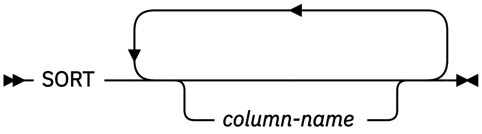
The following table lists primary commands that you can issue from any table display panel. For more information about these panels, see "Table display panels" on page 193.

Command	Alias	Description
BROWSE	B, BR, BRO, BROW	Browses the current ISPF table. For more information about BROWSE, see "BROWSE and VIEW panels" on page 194.

Command	Alias	Description

FIND		Finds a string on a table display panel.
		This command allows you to go directly to a particular string without having to scroll. The command starts at the row on which the cursor is positioned and searches all the columns, or the specified columns, for the specified string. If the string is found, the cursor is placed at the row.
		Syntax:
		
		string
		The search string. If the string contains special characters, enclose it in quotation marks (").
		*****
		Search for the last string used.
		NEXT
		Searches forward from the current location.
		PREV
		Searches backward from the current location.
		fromcolno
		The number of the column from which the search is to start. Specifying *fromcolno* limits the search to the specified column and the columns after it.
		tocolno
		The number of the column at which the search is to end. Specifying *tocolno* limits the search to the specified column and the columns before it.
		To find the column numbers for *fromcolno* and *tocolno*, issue the SORT command without any parameters.
		You can specify RFIND to repeat the last FIND command.
		Examples:
		`FIND MYUSERID`
		`FIND MYUSERID PREV`
		`FIND MYUSERID 2 4`
		`FIND MYUSERID PREV 2 4`

ORDER		Defines or modifies the ORDER BY clause in the SQL statement that retrieves data for the panel. This command displays the **Order By Columns (ADB2SORT)** panel, where you can specify the ORDER BY criteria.
REFRESH	REF	Refreshes the current ISPF table with data from Db2.

Command	Alias	Description
SAVE TABLE AS		Saves the current ISPF table with the specified name in the specified library. **Syntax:**  ***name*** The name under which the table is to be saved. **ddname** The library DD name. The data set that is associated with the DD name must be preallocated before you use this command. If you do not specify a library name, the default library ISPTABL is used.
SEARCH	SARG	Searches the current ISPF table. This command displays the **Search fields (ADB2SARG)** panel, which lists all of the columns in the current ISPF table. On this panel you can specify searches on individual columns by entering a search operator and a search value for the columns. See "Filtering data on panels" on page 198. **Syntax:**  **RESET** Clears your search specifications.
SORT		Sorts the current ISPF table based on the specified column or columns. **Syntax:**  ***column-name*** The column name. Instead of specifying a column name, you can place the cursor in the column that you want sorted. If you do not specify a column name, and the cursor is not in a column, Db2 Admin Tool displays a panel on which you can specify your sort criteria. See "Sorting display data" on page 202.
VIEW	VI	Opens a VIEW session for the current ISPF table. For more information about VIEW, see "BROWSE and VIEW panels" on page 194.

Primary commands for specific panels

The following table lists primary commands that are specific for certain panels.

Table 12. Db2 Admin Tool primary commands

Command	Alias	Description
ADD		Adds objects.
ADDC		Adds a check constraint.
ADDCOL		Adds a column to an index.
ADDU		Adds a unique or primary constraint.
AE		Displays all authorizations that are explicitly granted.
AI		Displays all authorizations that are held implicitly.
ALL		Lists all objects of a specified type for each object in a list of objects. For example, for a list of indexes on panel ADB21X, the ALL T command displays all tables that are associated with those indexes.
ALTOPT		Lets you specify options to control ALT processing.
AP		Displays all authorizations that are granted to PUBLIC.
ARCHIVE		Archives partitions from an accelerated table.
AU		Displays all authorizations that are held directly.
BALANCE		Sets the size for each partition to the average size.
BET		Generates a detail report in batch mode for tables and related objects, packages, and accelerated tables. The report displays the following types of information for tables and their related objects: • Table details • Column information • Index information • Information about keys • Information about aliases The package details report displays the following information: • Package details • SQL information • EXPLAIN information from the package owner's plan table This command is available on the **Tables, Views, and Aliases (ADB21T)** panel , the **Packages (ADB21K)** panel), and the **Display Accelerated Tables (ADBPZAT)** panel.
BIND		Generates BIND commands for multiple application packages or plans. The BIND commands are created in a work statement list. This command is valid only when packages or plans are displayed.
BINDOPT		Lets you specify bind and rebind options that are not in the Db2 catalog records.
BP		Change the batch parameters that are to be used.
CALL		Calls the procedure.
CAPTURE		Displays START DYNQUERYCAPTURE panel to stabilize cached dynamic SQL statements.

Table 12. Db2 Admin Tool primary commands (continued)

Command	Alias	Description
CHKNTS		Checks whether any of the new table spaces already exist when moving tables from multi-table table spaces to UTS.
CHKPDC		Checks whether any of the existing table spaces have pending changes.
CLEAR		Deletes all the rows.
COLUMNS		Performs a column lookup when primary, unique, or foreign key constraints are being added.
COMMENT		Toggles between displaying comments and hiding comments. COMMENT ON displays comments on a second line. COMMENT OFF hides comments. Entering COMMENT without ON or OFF toggles between the two settings.
COMP		Defines individual tracing components whose trace levels deviate from the default trace level.
CONSTRAINT S		Adds, alters, or drops a primary key, unique key, check constraint, or referential constraint for the table.
COPY	C	Runs the COPY utility on all objects in the list.
COPYALL	CA	Runs the COPY utility on all partitions in a single step.
CT		Migrates objects by using Db2 Cloning Tool.
DDEF		Displays Db2 definitions for the selected page sets.
DEL		Deletes all listed objects.
DET		Generates a detail report in online mode for tables and related objects, packages, and accelerated tables. The detail report contains the same information as the report for the BET command and is available from the same panels. **Restriction:** The DET primary command is available for only the following table types: • C: Clone table • G: Created global temporary table • H: History table • P: Implicit table that was created for XML columns • R: Archive table • T: Table • X: Auxiliary table
DIS		Generates a Db2 command to display information for all listed objects. This command is valid only when databases, table spaces, or indexes are displayed.
DISABLE		Disables the table for acceleration.
DISPINFO		Displays additional information (replication status and row count) for the table.
DISPOPT		Displays criteria to filter the list of accelerated tables.
DISPREF		Restores the original display on the **Display Accelerated Tables (ADBPZAT)** panel.

Table 12. Db2 Admin Tool primary commands (continued)

Command	Alias	Description
DROP		Drops all of the listed objects. You are prompted to confirm this DROP operation before it is executed.
DSTAT		Displays Db2 statistics for the selected page sets.
EDIT		Enables editing of the objects that are listed in a panel.
ENABLE		Enables the table for acceleration.
EXPLSTMTCA CHE		Extracts EXPLAIN records for all cached statements into the DSN_STATEMENT_CACHE_TABLE table.
EXPORT		Exports the multi-target change information to a file on the target system.
EXPRESSION		Creates a new key-expression. You can create multiple expressions. This command is available only the **Create Index (ADB21XAR)** panel.
EXTENTS		Displays the estimated number of extents needed.
EXTERNAL		Toggles to a view which includes the **External Name** column.
FREE		Generates FREE commands for multiple application packages or plans. The FREE commands are created in a work statement list. This command is valid only when packages or plans are displayed.
FREEALL		Frees all stabilized dynamic queries.
FREEINV		Frees all invalid stabilized dynamic queries.
FREEINVCOP Y		Frees only copies of stabilized dynamic queries that are invalid.
GEN		Generates SQL for the objects from the Db2 catalog.
GRANT		Generates a GRANT statement for all objects that are listed. The GRANT command is useful on authorization panels when copying authorizations from one user to one or more other users. This command is valid only when the values in the Grantee column are the same and one of the following types of objects are displayed: databases, tables, views, aliases, packages, plans, sequences, stored procedures, user-defined functions, user-defined data types, or authorizations.
HASH		Enables fast access to a row by hashing a key value and storing the hash value in a unique index.
HINT		Switches panel format to show hint IDs and hints used.
INCCOPY	CI	Creates an incremental copy (COPY FULL NO) of all objects in the list.
INCCOPYALL	CIA	Runs COPY FULL NO on all partitions in a single step.
INDEX		Switches panel format to show index name and schema.
JOBINFO		Displays JOBNAME and TIME STARTED values for utility jobs.
LASTPG		Displays the last logical page of statistics for the selected page sets.
LIKE		Switches the LIKE operator ON or OFF for search criteria. This command is valid only on the **System Catalog (ADB21)** panel.

Table 12. Db2 Admin Tool primary commands (continued)

Command	Alias	Description
LOAD		Loads or reloads data into a table.
LONGNAMES		Displays column names of 30 characters. Use the LONGNAMES and the SHORTNAMES commands to toggle between 30-character and 18-character column names on the panel.
MAKEPBG		Converts the table space to a parition-by-growth (PBG) table space. This command initializes input fields **Numparts**=0 and **Max Partitions**>0 to allow the conversion to a PBG table space.
MAKEPBR		Converts the table space to a parition-by-range (PBR) table space with absolute page numbering. This command initializes input fields **Numparts**>0, **Max Partitions**=0, and **SEGSIZE**>0 to allow the conversion to a PBR table space with absolute page numbering.
MAKEPBR2		Converts the table space to a parition-by-range (PBR) table space with relative page numbering. This command initializes input fields **Numparts**>0, **Max Partitions**=0, **SEGSIZE**>0, and **PAGENUM**=R to allow the conversion to a PBR table space with relative page numbering.
MIG		Migrates the displayed objects. This command is valid only when databases, table spaces, or tables are displayed.
MOVETB		Moves tables from multi-table table spaces to partition-by-growth (PBG) universal table spaces (UTS).
NEXT		Move to the next panel or generate a batch job.
NEXTCOL		Saves changes for the current column and moves to the next updatable table column.
OBJAUTH		Switch panel format to show system object authorizations.
ORIGINAL		Restores values to their existing state.
PACKAGES		List the packages that are affected by moving the tables.
PLANMGMT		Displays the plan management attributes for the packages.
QUALIFIER		Displays the qualifier for the packages.
RE-SORT		Re-sorts the table to its original sequence.
REBIND		Generates REBIND commands for multiple application packages or plans. The REBIND commands are created in a work statement list and contain only the package or plan name. This command is valid only when packages or plans are displayed. When you specify REBIND, the resulting BIND commands contain only the package or plan name. Specify REBIND FULL to have the resulting BIND commands contain both the package or plan name and all of the parameters.
REFRTS		Retrieves current real-time statistics for all objects in the list.
REORG	O	Runs the REORG utility on all objects in the list.
REORGALL	OA	Runs the REORG utility on all partitions in a single step.

Table 12. Db2 Admin Tool primary commands (continued)

Command	Alias	Description
REP		Generates a batch job that produces a printable report of the objects in the Db2 catalog.
RESET		Clears or deselects the list.
RESTORE		Restores objects or settings. When RESTORE is issued on the **Restore Partitions (ADBP1ARC)** panel, the selected archived partitions are restored.
RESZ		Generates jobs to resize page sets.
REVOKE		Generates REVOKE statements for all of the system authorities, user authorities, and object authorizations that are listed for the specified grantees. When you issue the REVOKE command, you are prompted to confirm that you want to execute the command because of the significant impact of the REVOKE command.
RMIMPL		Removes rows that represent implicit grants so that only explicitly granted authorizations are displayed. Implicit grants are grants where **Grantor** is the same as **Grantee** or **GT** (Grantee type) is P. To restore the display to include the implicit grants again, use the REFRESH command.
RO		Restore packages using the original DBRM.
ROTATE		Rotates a partition.
RR		Restore packages using the regenerated DBRM.
RTS		Displays real-time statistics for the tables listed on the panel.
RTSO		Displays real-time statistics for the tables listed on the panel using the user-defined options.
RUNSTATS	R	Runs the RUNSTATS utility on all objects in the list.
RUNSTATSALL	RA	Runs the RUNSTATS utility on all partitions in a single step.
SHORTNAMES		Displays column names of 18 characters. Use the LONGNAMES and the SHORTNAMES commands to toggle between 30-character and 18-character column names on the panel.
SPACE		Shows the amount of space (in KB) that is used for the VSAM page set.
STA		Generates a Db2 command to start all listed objects. This command is valid only when databases, table spaces, or indexes are displayed.
STARTTRACE		Starts a performance trace with IFCID 316, 317 or 318 and enable extraction of EXPLAIN records for all cached statements.
STO		Generates a Db2 command to stop all listed objects. This command is valid only when databases, table spaces, or indexes are displayed.
STOPGROUP		Stops all active capture monitors on all members of the data sharing group.
STOPLOCAL		Stops all active capture monitors on the local subsystem.
SYSAUTH		Switch panel format to show system authorizations.

Table 12. Db2 Admin Tool primary commands (continued)

Command	Alias	Description
TABLE		Switches panel format to show table name and schema.
TBLOPTS		Allows you to modify additional table attributes and specify period definitions for the table. You make these modifications on the **Alter - Table Options (ADBP7TOP)** panel.
TU		Changes the use of Db2 templates.
WSLOPT		Set work statement list (WSL) options.
UTIL		Generates utility JCL for the table spaces in all of the listed databases.
UO		Modifies utility options.
VALUES		Displays current LIMITKEY values for each partition.
VDEF		Displays VSAM definitions for the selected page sets.
VERSION		Toggles to a view which includes the **Version** and **A** (Active) columns
VERSIONS		Displays the package version, bind timestamp, and consistency token.
VSTAT		Displays VSAM statistics.
XSPACE		Displays the index space names.
ZOOM		Collapses or expands a section or all sections.

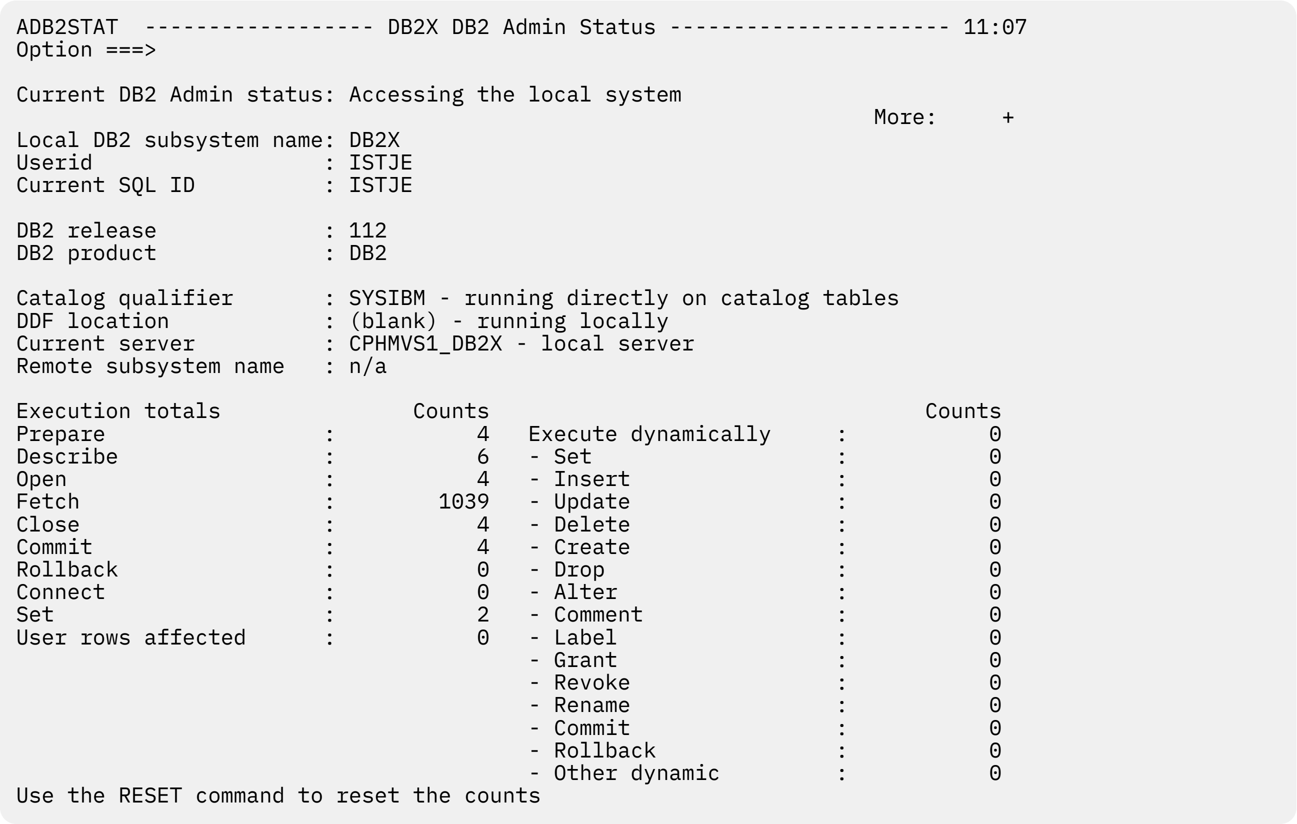
Checking the status of Db2 Admin Tool

You can display and verify certain information about the current Db2 Admin Tool session, such as the current SQL ID, local subsystem name, and the execution count for specific operations (such as COMMIT and SET).

Procedure

From any panel in Db2 Admin Tool, issue the STATUS primary command.

The **DB2 Admin Status (ADB2STAT)** panel is displayed:



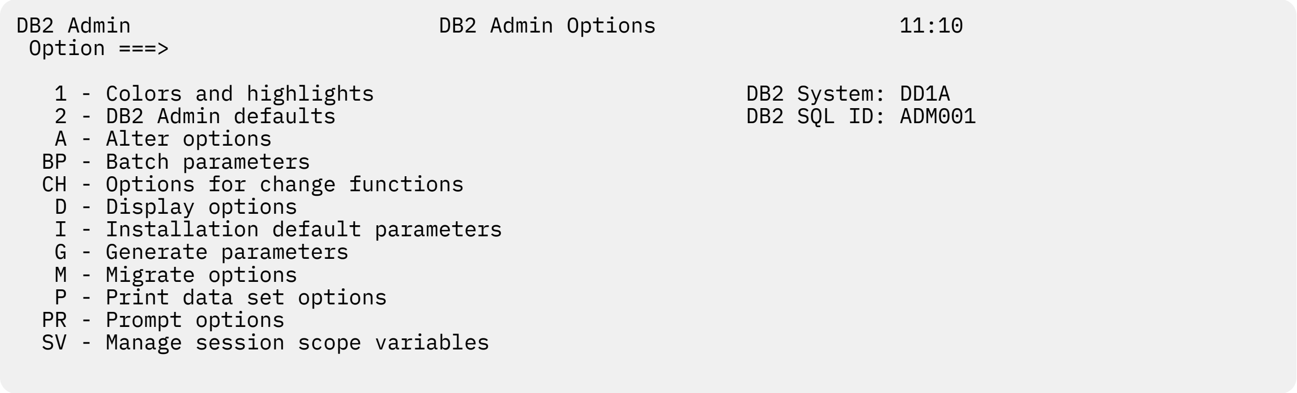
```
ADB2STAT  ------------------ DB2X DB2 Admin Status ---------------------- 11:07
Option ===>

Current DB2 Admin status: Accessing the local system
                                                                    More:     +
Local DB2 subsystem name: DB2X
Userid                  : ISTJE
Current SQL ID          : ISTJE

DB2 release             : 112
DB2 product             : DB2

Catalog qualifier       : SYSIBM - running directly on catalog tables
DDF location            : (blank) - running locally
Current server          : CPHMVS1_DB2X - local server
Remote subsystem name   : n/a

Execution totals              Counts                                 Counts
Prepare                 :          4   Execute dynamically     :          0
Describe                :          6   - Set                   :          0
Open                    :          4   - Insert                :          0
Fetch                   :       1039   - Update                :          0
Close                   :          4   - Delete                :          0
Commit                  :          4   - Create                :          0
Rollback                :          0   - Drop                  :          0
Connect                 :          0   - Alter                 :          0
Set                     :          2   - Comment               :          0
User rows affected      :          0   - Label                 :          0
                                       - Grant                 :          0
                                       - Revoke                :          0
                                       - Rename                :          0
                                       - Commit                :          0
                                       - Rollback              :          0
                                       - Other dynamic         :          0
Use the RESET command to reset the counts
```

*Figure 98. **DB2 Admin Status (ADB2STAT)** panel*

Settings in Db2 Admin Tool

In Db2 Admin Tool, you can specify preferences and default settings, such as color schemes, prompt options, and options for change functions.

Use the **DB2 Admin Options (ADB2P)** panel (option P on the main menu) to change these settings.



```
DB2 Admin                       DB2 Admin Options                 11:10
 Option ===>

   1 - Colors and highlights                          DB2 System: DD1A
   2 - DB2 Admin defaults                             DB2 SQL ID: ADM001
   A - Alter options
  BP - Batch parameters
  CH - Options for change functions
   D - Display options
   I - Installation default parameters
   G - Generate parameters
   M - Migrate options
   P - Print data set options
  PR - Prompt options
  SV - Manage session scope variables
```

*Figure 99. **DB2 Admin Options (ADB2P)** panel*

Changing colors and highlights

You can change the colors or highlighting scheme (or designations) technique on Db2 Admin Tool panels.

Procedure

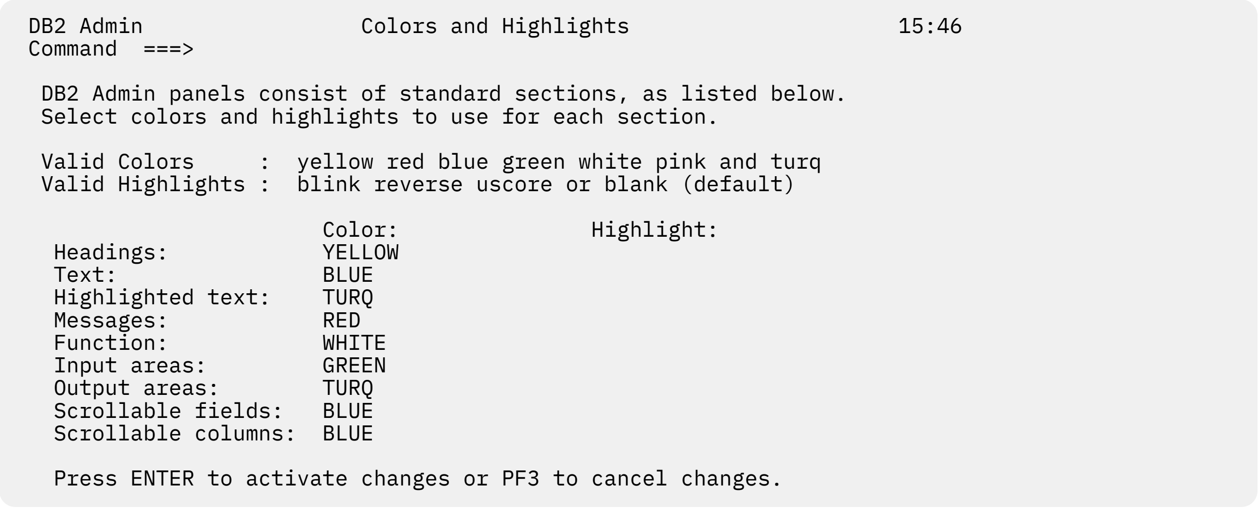
1. On the **DB2 Administration Menu (ADB2)** panel, specify option P, and press Enter.

   Alternatively, issue the OPTIONS primary command on any Db2 Admin Tool panel.

The **DB2 Admin Options (ADB2P)** panel is displayed.

2. Specify option 1, and press Enter.

   The **Colors and Highlight (ADB2P1)** panel is displayed, as shown in the following figure:



```
DB2 Admin                 Colors and Highlights                     15:46
Command  ===>

 DB2 Admin panels consist of standard sections, as listed below.
 Select colors and highlights to use for each section.

 Valid Colors     :  yellow red blue green white pink and turq
 Valid Highlights :  blink reverse uscore or blank (default)

                      Color:               Highlight:
  Headings:           YELLOW
  Text:               BLUE
  Highlighted text:   TURQ
  Messages:           RED
  Function:           WHITE
  Input areas:        GREEN
  Output areas:       TURQ
  Scrollable fields:  BLUE
  Scrollable columns: BLUE

  Press ENTER to activate changes or PF3 to cancel changes.
```

*Figure 100. **Colors and Highlight (ADB2P1)** panel*

3. Change the colors or highlighting schemes of the following sections on the panels, and press Enter.

   Headings
   The first line of the panel. The default setting is yellow.

   Text
   The instructions or descriptions on the panel. The default setting is blue.

   Highlighted text
   The emphasized text. The default setting is turquoise.

   Messages
   The message area, or the third line on the panel when a message is returned. The default setting is red.

   Function
   The command line, the option chosen, or both. The default setting is white.

   Input areas
   The areas in which you specify your input. The default setting is green.

   Output areas
   The areas in which output is returned. The default setting is turquoise.

   Scrollable fields
   The fields that you can scroll for more information. The default setting is blue.

   Scrollable columns
   The columns that you can scroll for more information. The default setting is blue.

   If you leave an input field on the panel blank, the default value is used. To choose default values for all sections of the panel, specify RESET on the command line, and press Enter.

Changing defaults

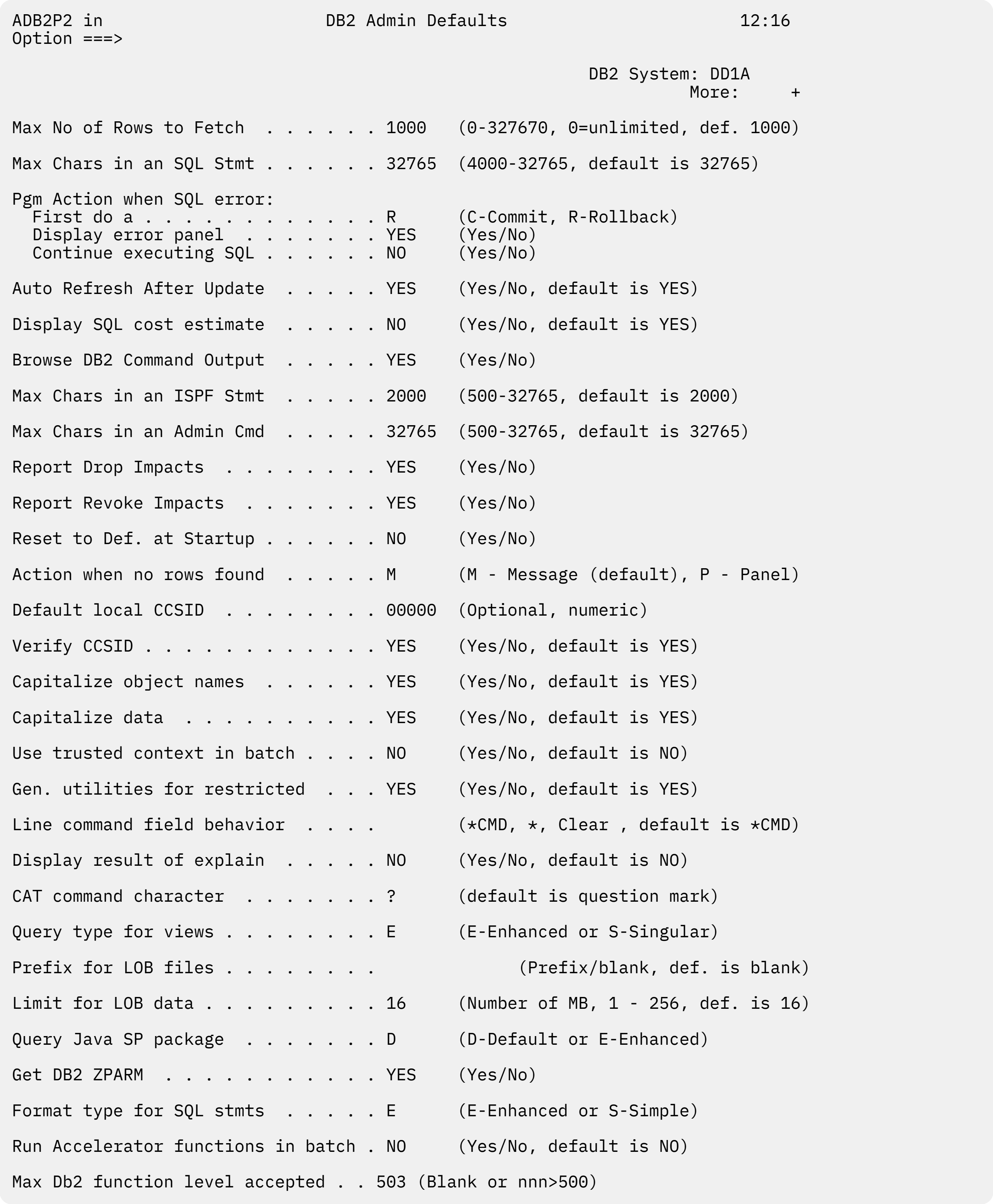
You can change various parameters that affect the execution of Db2 Admin Tool by using the **Admin Defaults (ADB2P2)** panel.

Procedure

To change defaults in Db2 Admin Tool:

1. Issue the OPTIONS primary command on any Db2 Admin Tool panel.

2. On the **DB2 Admin Options (ADB2P)** panel, specify option 2, and press Enter.
3. On the **DB2 Admin Defaults (ADB2P2)** panel, edit the values as needed. For information about each field, press PF1 to access the help.



```
ADB2P2 in                      DB2 Admin Defaults                      12:16
Option ===>

                                                         DB2 System: DD1A
                                                                   More:     +

Max No of Rows to Fetch  . . . . . . 1000   (0-327670, 0=unlimited, def. 1000)

Max Chars in an SQL Stmt . . . . . . 32765  (4000-32765, default is 32765)

Pgm Action when SQL error:
  First do a . . . . . . . . . . . . R      (C-Commit, R-Rollback)
  Display error panel  . . . . . . . YES    (Yes/No)
  Continue executing SQL . . . . . . NO     (Yes/No)

Auto Refresh After Update  . . . . . YES    (Yes/No, default is YES)

Display SQL cost estimate  . . . . . NO     (Yes/No, default is YES)

Browse DB2 Command Output  . . . . . YES    (Yes/No)

Max Chars in an ISPF Stmt  . . . . . 2000   (500-32765, default is 2000)

Max Chars in an Admin Cmd  . . . . . 32765  (500-32765, default is 32765)

Report Drop Impacts  . . . . . . . . YES    (Yes/No)

Report Revoke Impacts  . . . . . . . YES    (Yes/No)

Reset to Def. at Startup . . . . . . NO     (Yes/No)

Action when no rows found  . . . . . M      (M - Message (default), P - Panel)

Default local CCSID  . . . . . . . . 00000  (Optional, numeric)

Verify CCSID . . . . . . . . . . . . YES    (Yes/No, default is YES)

Capitalize object names  . . . . . . YES    (Yes/No, default is YES)

Capitalize data  . . . . . . . . . . YES    (Yes/No, default is YES)

Use trusted context in batch . . . . NO     (Yes/No, default is NO)

Gen. utilities for restricted  . . . YES    (Yes/No, default is YES)

Line command field behavior  . . . .        (*CMD, *, Clear , default is *CMD)

Display result of explain  . . . . . NO     (Yes/No, default is NO)

CAT command character  . . . . . . . ?      (default is question mark)

Query type for views . . . . . . . . E      (E-Enhanced or S-Singular)

Prefix for LOB files . . . . . . . .              (Prefix/blank, def. is blank)

Limit for LOB data . . . . . . . . . 16     (Number of MB, 1 - 256, def. is 16)

Query Java SP package  . . . . . . . D      (D-Default or E-Enhanced)

Get DB2 ZPARM  . . . . . . . . . . . YES    (Yes/No)

Format type for SQL stmts  . . . . . E      (E-Enhanced or S-Simple)

Run Accelerator functions in batch . NO     (Yes/No, default is NO)

Max Db2 function level accepted . . 503 (Blank or nnn>500)
```

*Figure 101. **DB2 Admin Defaults (ADB2P2)** panel*

DB2 Admin Defaults (ADB2P2) panel has the following fields:

Max No of Rows to Fetch
Specifies the maximum number of rows to fetch for each SQL SELECT statement. The default value is 1000. If you do not want to limit the number of rows that are fetched, specify 0. In this case, the entire result set is fetched. However, be aware that specifying 0 or a large number can result in long response times for queries that have a large result set.

Max Chars in an SQL Stmt
Specifies the maximum length of the buffer for SQL statements. Db2 Admin Tool allocates the specified number of bytes when displaying a new panel. Specifying a high value can slow TSO performance on a storage-constrained system.

Pgm Action when SQL error
Specifies the action that Db2 Admin Tool takes when an SQL error occurs.

First do a
Indicates whether Db2 Admin Tool performs a COMMIT or ROLLBACK operation.

Display error panel
Indicates whether Db2 Admin Tool displays the SQL error panel with the SQL error message and SQLCA.

Continue executing SQL
Indicates whether to continue processing and execute the next SQL statement

Auto Refresh After Update
Indicates whether table display panels are to be refreshed after SQL updates.

If YES is specified, Db2 Admin Tool refreshes the panels when they are displayed again. For performance reasons, this refresh is limited to panels where the elapsed time to fetch the rows to be displayed is less than 10 seconds.

If NO is specified, panels are not refreshed. This situation might result in you viewing and acting on old data when you press END.

Display SQL cost estimate
Indicates whether Db2 Admin Tool displays an estimated cost for an SQL SELECT statement. This estimate is displayed as an ISPF message. If the estimated cost is larger than the maximum value of an integer, the estimated cost is displayed as "*.***.***.***".

Browse DB2 Command Output
Indicates whether Db2 Admin Tool invokes ISPF browse (YES) or lets the output default to TSO line mode (NO).

Max Chars in an ISPF Stmt
Specifies the maximum length of the buffer for ISPF statements. Specifying a large number can slow TSO performance on a storage-constrained system.

Max Chars in an Admin Cmd
Specifies the maximum length of the buffer for Db2 Admin Tool commands. Specifying a large number can slow TSO performance on a storage-constrained system.

Report Drop Impacts
Specifies the default value to be displayed in the **Report Drop Impacts** field when dropping an object.

Report Revoke Impacts
Specifies the default value to be displayed in the **Report Revoke Impacts** field when revoking authorities.

Reset to Def. at Startup
Indicates whether Db2 Admin Tool restores the following fields to their default values at the next startup:

- MAX NO OF ROWS TO FETCH
- MAX CHARS IN AN SQL STATEMENT
- AUTO REFRESH AFTER UPDATE

- MAX CHARS IN AN ISPF STMT
- MAX CHARS IN AN ADMIN CMD

If NO is specified, Db2 Admin Tool attempts to restore the CURRENT SQLID.

Action when no rows found
Indicates whether Db2 Admin Tool displays a panel or a message when no rows are found.

Default local CCSID
Specifies a default CCSID value to use if no terminal CCSID is available in ISPF variable ZTERMCID. This default CCSID value is used to enable the SQ line command for packages, plans, and triggers.

Verify CCSID
Indicates whether Db2 Admin Tool verifies that the CCSID for the TSO terminal matches the CCSID for the plan under which Db2 Admin Tool is running. When you start Db2 Admin Tool or Object Comparison Tool and verification is active, a warning is displayed if the CCSIDs do not match. Different CCSIDs can cause unexpected data conversions for any characters that do not map to the same code point in the two CCSIDs.

Capitalize object names
Indicates whether Db2 Admin Tool translates the lowercase characters to uppercase characters in object names, qualifiers, and authorization identifiers in the following fields on the **System Catalog (ADB21)** panel:

- Name
- Owner
- In DB/Coll (databases and collections)
- Grantor
- Grantee

If NO is specified, Db2 Admin Tool supports the use of lowercase characters in the qualifier and name of the object when you use Db2 Admin Tool panels to:

- Create or drop an index.
- Create or drop an view.
- Drop a table.

Some Db2 object names are required to have only uppercase characters, such as database names, table space names, plan names, and package names (except for trigger package names). Db2 Admin Tool always translates any lowercase characters in these objects names to uppercase characters, even if NO is specified in the **Capitalize object names** field.

Tip: Use the CAPS primary command to temporarily override the **Capitalize object names** setting on the catalog navigation panels.

Capitalize data
Indicates whether Db2 Admin Tool translates lowercase characters to uppercase characters for data that you enter. For example, if NO is specified, comment fields on Db2 Admin Tool panels remain in the case in which they were entered.

This field does not apply to objects that are managed by Db2 Admin Tool.

Use trusted context in batch
Indicates whether the ASUSER parameter that is specified when Db2 Admin Tool starts should also be used in batch. The job name in the jobs that are submitted must match the job name in the trusted context.

Gen. utilities for restricted
Indicates whether Db2 Admin Tool should prompt for additional utilities when Db2 places an object in an restrictive state. (Db2 returns SQLCODE +610 to indicate that an object was placed in a restrictive state.)

Line command field behavior
Specifies what to display in the line command field after a line command is processed.

***CMD**

Specifies that an asterisk is to be displayed in the first position of the line command.

*****

Specifies that only an asterisk is to be displayed in the line command field. (The line command is cleared.)

CLEAR

Specifies that the line command is to be cleared and nothing is displayed in the line command field.

Display result of explain

Indicates whether Db2 Admin Tool displays the EXPLAIN information if CURRENT EXPLAIN MODE is YES or EXPLAIN. (When CURRENT EXPLAIN MODE is EXPLAIN, Db2 does not return any data, only the EXPLAIN information.)

CAT command character

Specifies a character that can be used as a shortcut for the CAT command. The character cannot be alphanumeric, the current value of the ISPF command delimiter, or any of the following characters:

+ (plus)
- (minus)
% (percentage)
& (ampersand)
= (equal)
< (less than)
> (greater than)
$ (dollar sign)
# (pound sign)
@ (at symbol)
{ (left bracket)
¦ (pipe)
} (right bracket)
_ (underscore)
' (single quote)
: (colon)
" (double quote)

Query type for views

Specifies the type of query to use to get information for views.

S

Singular query. The output includes only the contents of SYSTABLES.

E

Enhanced query. The output includes more information that is presented in a view-centric display. If you select E, you can use COLUMN search on columns in SYSVIEWS.

Prefix for LOB files

Specifies the high level qualifier(s) for LOB files. The default is blank. If the prefix contains a period, the TSO prefix is not appended to the file name.

Limit for LOB data

Specifies the maximum amount of LOB data to fetch (in MB) when LOB data is stored in ISPF tables.

Query Java™ SP package

Specifies the algorithm to use for locating the packages of a Java stored procedure, when the K line command is issued on the**Stored Procedures (ADB21O)** panel.

D

Packages are located by using the COLLID value and EXTERNAL NAME value of the Java stored procedure, which are stored in Db2 catalog tables. D is the default.

E

Packages are located by using the default algorithm with the following additions:

- If no packages are found, the Db2 Admin Tool attempts to locate packages by using the COLLID value and CLASS value of the stored procedure.
- If CLASS is embedded in the REMARKS column of a package and one of the following conditions are true, Db2 Admin Tool associates the package with the stored procedure:
  - The COLLID value of the package is equal to the COLLID value of the stored procedure.
  - The COLLID value of the package is NULLID if the COLLID value of the stored procedure is blank.

Get DB2 ZPARM

Indicates whether to call ADMIN_INFO_SYSPARM during a Db2 Admin Tool process to get Db2 subsystem parameters. The default value is YES.

ADMIN_INFO_SYSPARM requires Db2 monitor privileges. Db2 Admin Tool does not call ADMIN_INFO_SYSPARM during the bind or rebind process.

Format type for SQL stmts

Specifies the format for displaying SQL statements when displaying package details. (You display package details by using the DET line command on the **Packages (ADB21K)** panel.)

S

Displays SQL statements in simple format, with 72 bytes of text per line and host variable information on additional lines.

E

Displays SQL statements with complex nested subqueries in enhanced format, where the queries are formatted for readability. Enhanced format applies to only DECLARE CURSOR and SELECT statements. All other statements are displayed in simple format.

Run Accelerator functions in batch

Specifies whether eligible accelerator functions are run in batch (YES) or in TSO (NO).

Max Db2 function level accepted

The maximum function level on which you want to allow Db2 Admin Tool to run, even if that function level is not tolerated or supported.

Changing alter options

Use the **Alter Options** panel to change settings for the ALTER command.

Procedure

On the **DB2 Admin Options** panel, specify option A, and press Enter.

The **Alter Options** panel is displayed.

Changing batch parameters

Use the **Batch Job Utility Parameters** panel to change batch job settings.

Procedure

On the **DB2 Admin Options** panel, specify option BP, and press Enter.

The **Batch Job Utility Parameters** panel is displayed.

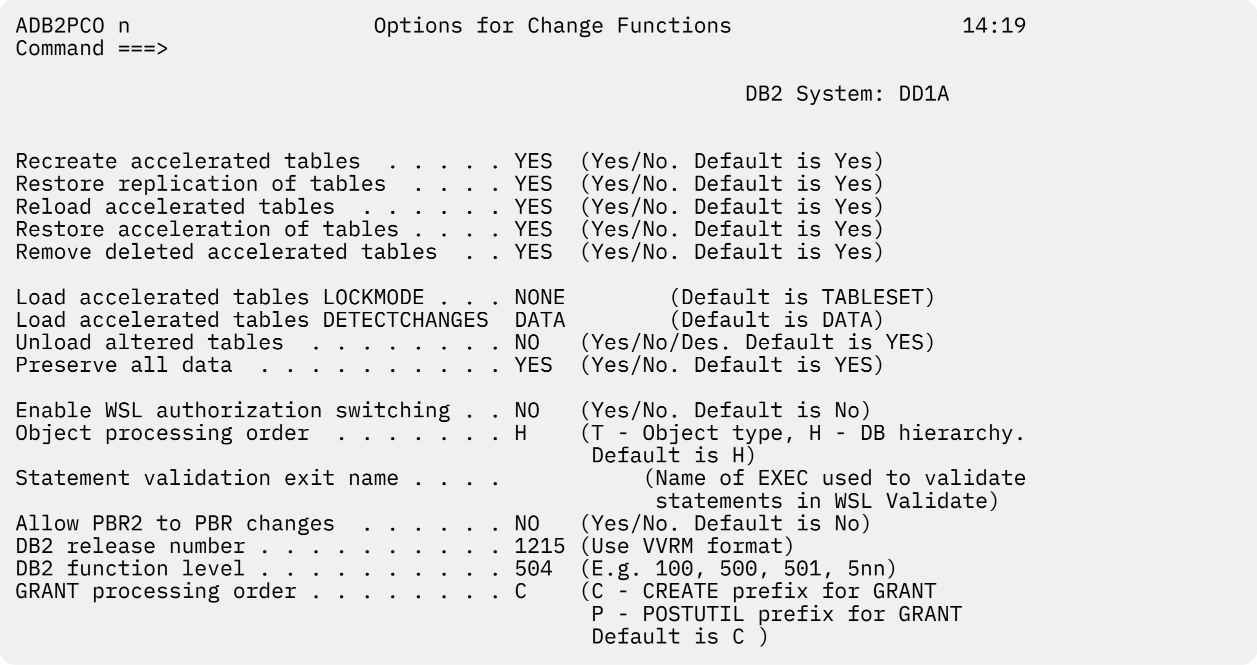
Changing options for change functions

You can specify certain settings that are common to change functions in Db2 Admin Tool, such as ALT and Compare. For example, you can specify whether to recreate accelerated tables, whether to unload tables during the analyze step, and the order in which objects are processed. These settings and more are listed on the **Options for Change Functions (ADB2PCO)** panel.

Procedure

To change these options:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option P, and press Enter.
2. On the **DB2 Admin Options (ADB2P)** panel, specify option CH, and press Enter.
3. On the **Options for Change Functions (ADB2PCO)** panel, change the values as needed. For detailed information about each field, see the help (PF1).



```
ADB2PCO n                   Options for Change Functions                  14:19
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 LOCKMODE . . . NONE        (Default is TABLESET)
Load accelerated tables DETECTCHANGES  DATA        (Default is DATA)
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 . . . . . . . . . . 1215 (Use VVRM format)
DB2 function level . . . . . . . . . . 504  (E.g. 100, 500, 501, 5nn)
GRANT processing order . . . . . . . . C    (C - CREATE prefix for GRANT
                                             P - POSTUTIL prefix for GRANT
                                             Default is C )
```

*Figure 102. **Options for Change Functions (ADB2PCO)** panel*

Changing display options

Use the **Display** panel to customize the display for supported table display panels.

Procedure

On the **DB2 Admin Options** panel, specify option D, and press Enter.

The **Panel Display Options** panel is displayed.

Specifying global PARALLEL values for utilities

You can set the values of the PARALLEL parameter for utilities that are run from Db2 Admin Tool.

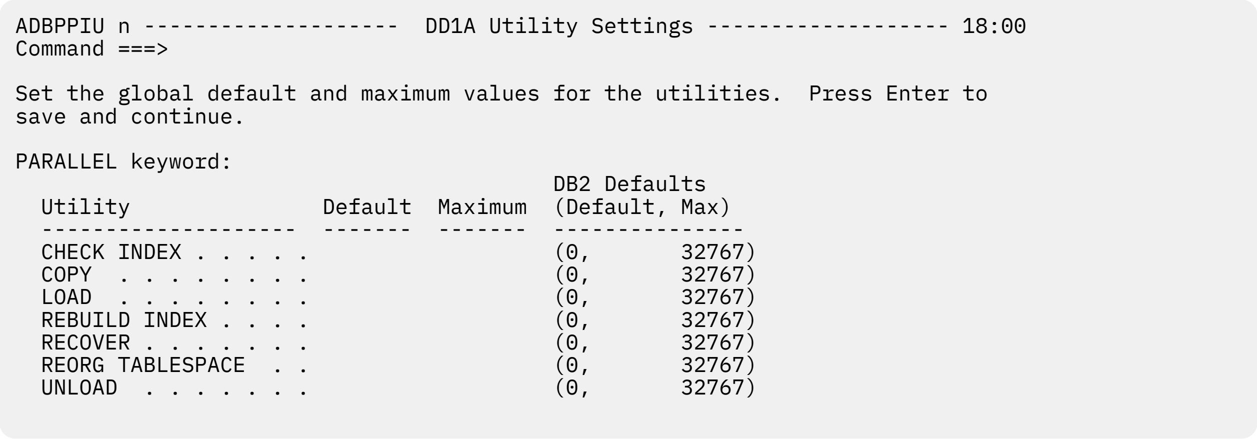
Procedure

To specify global PARALLEL values for utilities:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option P, and press Enter.

   Alternatively, issue the OPTIONS primary command on any Db2 Admin Tool panel.

2. On the **DB2 Admin Options (ADB2P)** panel, specify option I, and press Enter.
3. On the **DB2 Admin Installation Defaults (ADBPPI)** panel, option 1, and press Enter.
4. On the **Utility Settings (ADBPPIU)** panel, specify the default and maximum values for the PARALLEL keyword for the listed utilities, and press Enter:



```
ADBPPIU n --------------------  DD1A Utility Settings ------------------- 18:00
Command ===>

Set the global default and maximum values for the utilities.  Press Enter to
save and continue.

PARALLEL keyword:
                                          DB2 Defaults
  Utility              Default  Maximum  (Default, Max)
  --------------------  -------  -------  ---------------
  CHECK INDEX . . . . .                   (0,       32767)
  COPY  . . . . . . . .                   (0,       32767)
  LOAD  . . . . . . . .                   (0,       32767)
  REBUILD INDEX . . . .                   (0,       32767)
  RECOVER . . . . . . .                   (0,       32767)
  REORG TABLESPACE  . .                   (0,       32767)
  UNLOAD  . . . . . . .                   (0,       32767)
```

*Figure 103. **Utility Settings (ADBPPIU)** panel*

Generating parameters

Use the **Generating parameters** panel to manage the Generate function.

Procedure

On the **DB2 Admin Options** panel, specify option G, and press Enter.

The **Additional Generate Parameters** panel is displayed.

Changing migrate settings

Use the migrate function to change the parameter that controls whether space information is gathered and displayed in the **Migrate Table Spaces** panel (ADB28S).

Procedure

1. On the **DB2 Admin Options** panel, specify option M, and press Enter.

   The **Change Migrate Settings** panel is displayed.
2. Specify YES or NO in the **Show space information on panels** field.

Changing the SQL ID

Db2 uses the current SQL ID for the CREATE, GRANT, and REVOKE SQL statements.

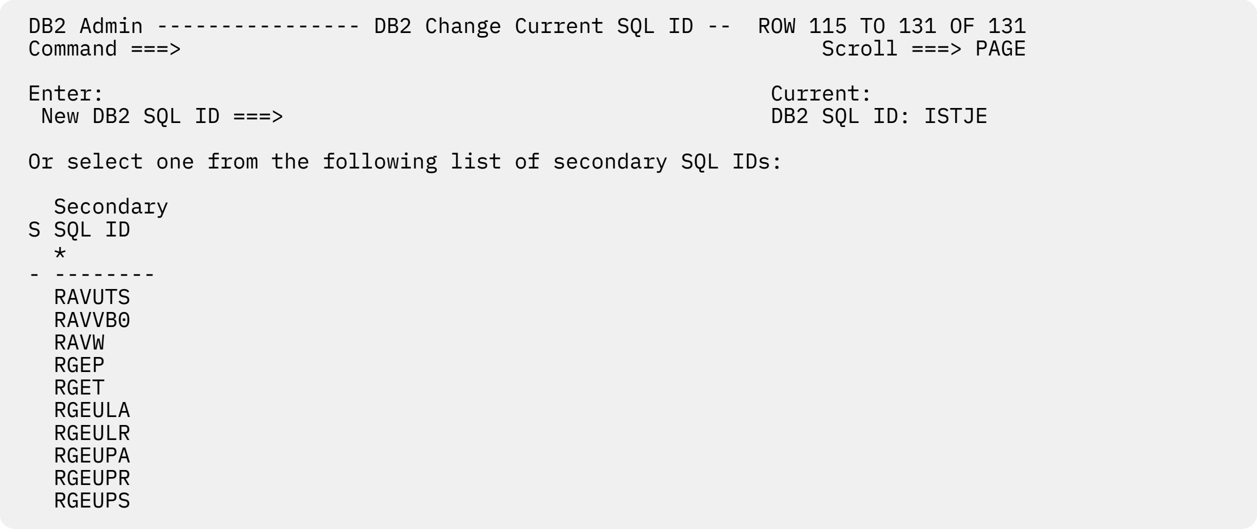
About this task

In all other cases, Db2 uses the composite privileges, that is, the combined privileges of your current, primary, and secondary SQL IDs.

Procedure

1. Specify option 4 on the **Administration Menu** panel, and press Enter.

   The **Change Current SQL ID** panel is displayed, as shown in the following figure.



```
DB2 Admin ---------------- DB2 Change Current SQL ID --  ROW 115 TO 131 OF 131
Command ===>                                                  Scroll ===> PAGE

Enter:                                                    Current:
 New DB2 SQL ID ===>                                      DB2 SQL ID: ISTJE

Or select one from the following list of secondary SQL IDs:

  Secondary
S SQL ID
  *
- --------
  RAVUTS
  RAVVB0
  RAVW
  RGEP
  RGET
  RGEULA
  RGEULR
  RGEUPA
  RGEUPR
  RGEUPS
```

*Figure 104. **Change Current SQL ID** panel (ADB24)*

2. Either enter a new SQL ID in the **New DB2 SQL ID** field or use the S line command to select from the list of secondary SQL IDs.

   The list of secondary SQL IDs is created by simulating or invoking the authorization exit in your system. The SET CURRENT SQLID='sqlidname' command is issued to change the current SQL ID.

   Requirement: To change the current SQL ID to one that is not included in the list of secondary SQL IDs, you must have SYSADM privilege.

   The ID you choose remains in effect until you change it again.

Changing and allocating print data sets

Allocate a print data set for the Db2 Admin Tool print function.

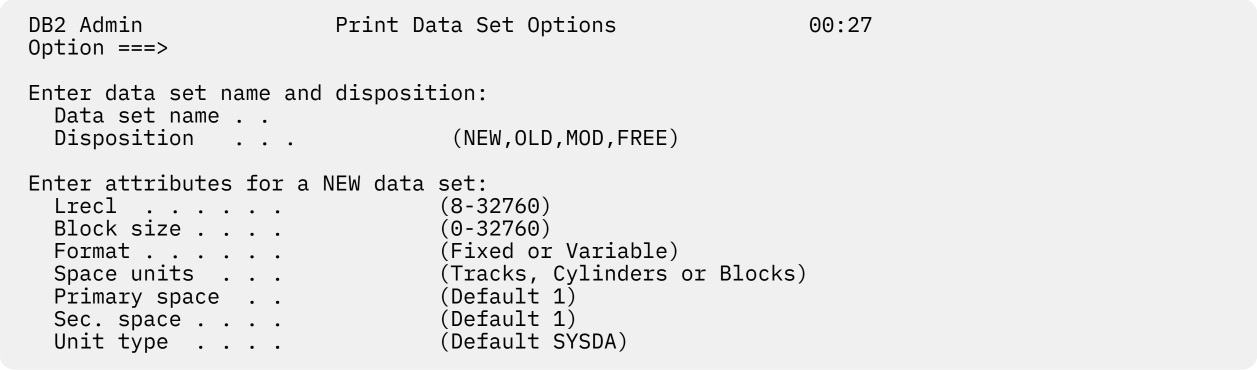
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option P, and press Enter.

   Alternatively, issue the OPTIONS primary command on any Db2 Admin Tool panel.

   The **DB2 Admin Options (ADB2P)** panel is displayed.
2. Specify option P, and press Enter.

   The **Print Data Set Options (ADB2PP)** panel is displayed, as shown in the following figure:



```
DB2 Admin               Print Data Set Options               00:27
Option ===>

Enter data set name and disposition:
  Data set name . .
  Disposition   . . .            (NEW,OLD,MOD,FREE)

Enter attributes for a NEW data set:
  Lrecl  . . . . . .            (8-32760)
  Block size . . . .            (0-32760)
  Format . . . . . .            (Fixed or Variable)
  Space units  . . .            (Tracks, Cylinders or Blocks)
  Primary space  . .            (Default 1)
  Sec. space . . . .            (Default 1)
  Unit type  . . . .            (Default SYSDA)
```

*Figure 105. **Print Data Set Options (ADB2PP)** panel*

   Specify the information for the print data set in the following fields:

   Data set name
      The name of the data set for Db2 Admin Tool to use for printing.

Disposition
One of the following allocation modes of the data set:

NEW
Allocates the new data set.

OLD
Uses an existing data set.

MOD
Appends output to an existing data set.

FREE
Deallocates the print data set.

If you want to allocate a new print data set, specify the following information:

Lrecl
The logical record length.

Block size
The block size.

Format
The data set format. Valid values are Fixed (F) and Variable (V).

Space units
The units in which space is to be allocated. Valid values are Tracks, Cylinders, and Blocks.

Primary space
The primary space allocation specified in preceding units.

Sec. space
The secondary space allocation specified in preceding units.

Unit type
The type of UNIT for allocation.

What to do next

To learn more about how to use the Db2 Admin Tool print function, review "Example: Printing ISPF table content to a data set" on page 240.

Example: Printing ISPF table content to a data set

You can use the Db2 Admin Tool print function to capture the contents of an ISPF table to a data set.

Step 1: Create the file that you want to send content to

Determine the format that you want for your data set based on the data that you want to store.

In this example, the data set name is `NEWONE.SAMPLE.PRINT`. NEWONE is the qualifier.

```
ISRUAIES DSLIST                       Data Set Information
Command ===>

Data Set Name . . . . : NEWONE.SAMPLE.PRINT

General Data                           Current Allocation
Management class . . : PRIMARY         Allocated cylinders : 1
Storage class  . . . : NORMALG         Allocated extents . : 1
 Volume serial . . . : SM4225
 Device type . . . . : 3390
Data class . . . . . : **None**
 Organization  . . . : PS             Current Utilization
 Record format . . . : FB              Used cylinders  . . : 0
 Record length . . . : 133             Used extents  . . . : 0
 Block size  . . . . : 27930
 1st extent cylinders: 1
 Secondary cylinders : 1              Dates
 Data set name type  :                 Creation date . . . : 2013/08/27
                                       Referenced date . . : ***None***
                                       Expiration date . . : ***None***
 SMS Compressible  . : NO
```

The following fields control the format of the data set:

Organization
Physically sequential (PS)

Record format
Fixed block (FB)

Record length
LRECL 133

Block size
BZSIZE 27930

Step 2: Allocate the data set in the PRINT data definition (DD)

You can allocate the data set in the PRINT DD or PRTTAB DD either through a logon procedure or the TSO ALLOC command. For example, you can run the following command: `TSO ALLOC F(PRINT) DSN('NEWONE.SAMPLE.PRINT') OLD`

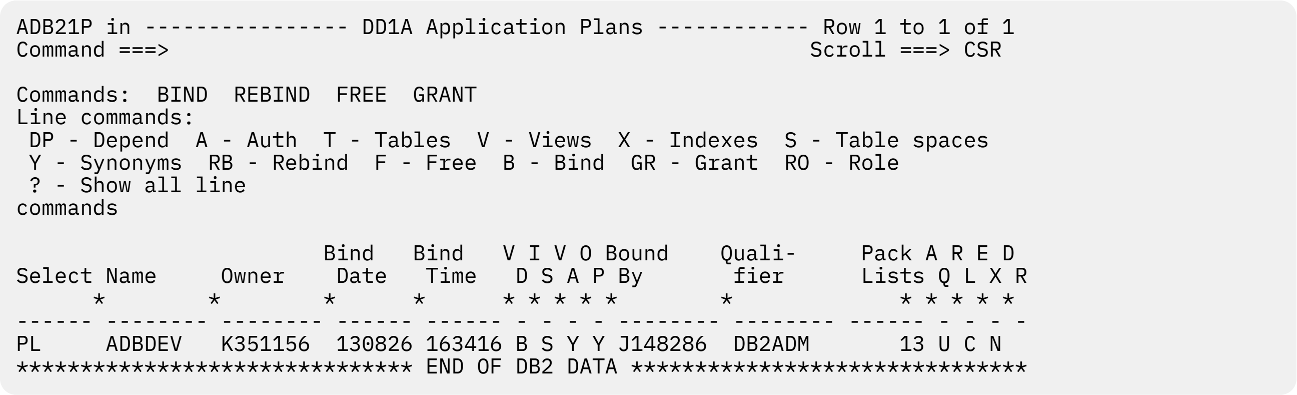
The print data set can also be allocated within Db2 Admin Tool by using the option P.P to access the following panel:

```
DB2 Admin -------------- Change/Allocate Print Data Set ----------------- 07:14
Option ===>

Enter data set name and disposition:
  Data set name ===> 'NEWONE.SAMPLE.PRINT'
  Disposition   ===> NEW        (NEW,OLD,MOD,FREE)

For a NEW data set enter:
  Lrecl         ===> 133        (8-32760)
  Block size    ===> 27930      (0-32760)
  Format        ===> F          (Fixed or Variable)
  Space units   ===> T          (Tracks, Cylinders or Blocks)
  Primary space ===>            (Default 1)
  Sec. space    ===>            (Default 1)
  Unit type     ===>            (Default SYSDA)
```

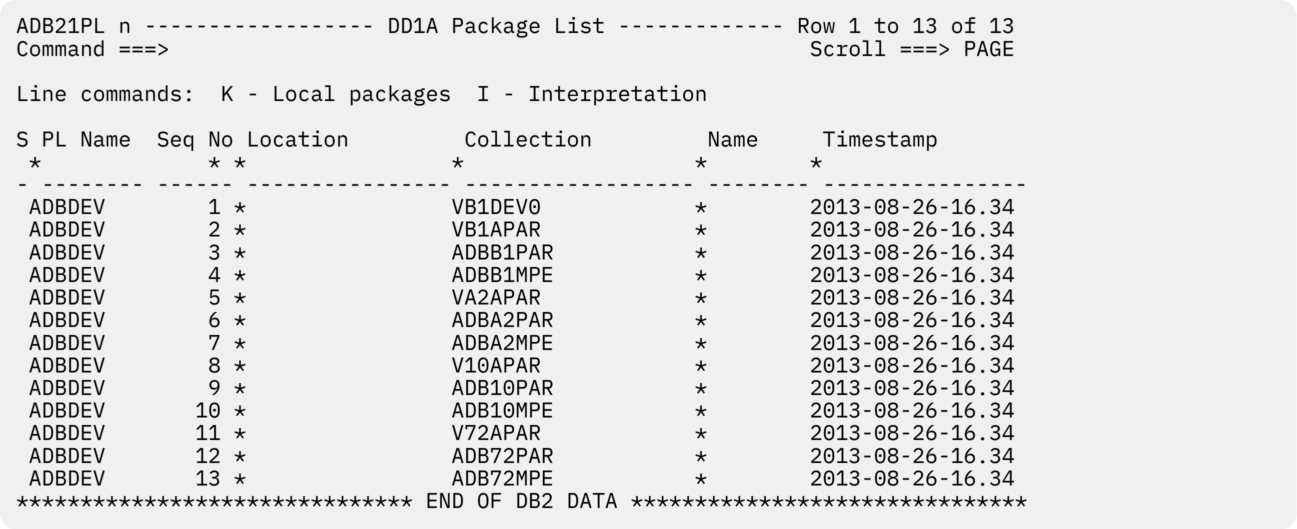
On this panel, you can allocate the data set to DD name PRINT in preparation for using the print command PRT TABLE ON FILE PRINT.

Step 3: View what you want to print

In this example, the content that is to be printed is a package list. In Db2 Admin Tool, navigate to the object that you want to print.



```
ADB21P in ---------------- DD1A Application Plans ------------ Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Commands:  BIND  REBIND  FREE  GRANT
Line commands:
 DP - Depend  A - Auth  T - Tables  V - Views  X - Indexes  S - Table spaces
 Y - Synonyms  RB - Rebind  F - Free  B - Bind  GR - Grant  RO - Role
 ? - Show all line
commands

                        Bind   Bind   V I V O Bound    Quali-     Pack A R E D
Select Name     Owner    Date   Time   D S A P By       fier      Lists Q L X R
      *        *        *      *      * * * * *        *             * * * * *
------ -------- -------- ------ ------ - - - - -------- -------- ------ - - - -
PL     ADBDEV   K351156  130826 163416 B S Y Y J148286  DB2ADM       13 U C N
******************************* END OF DB2 DATA *******************************
```

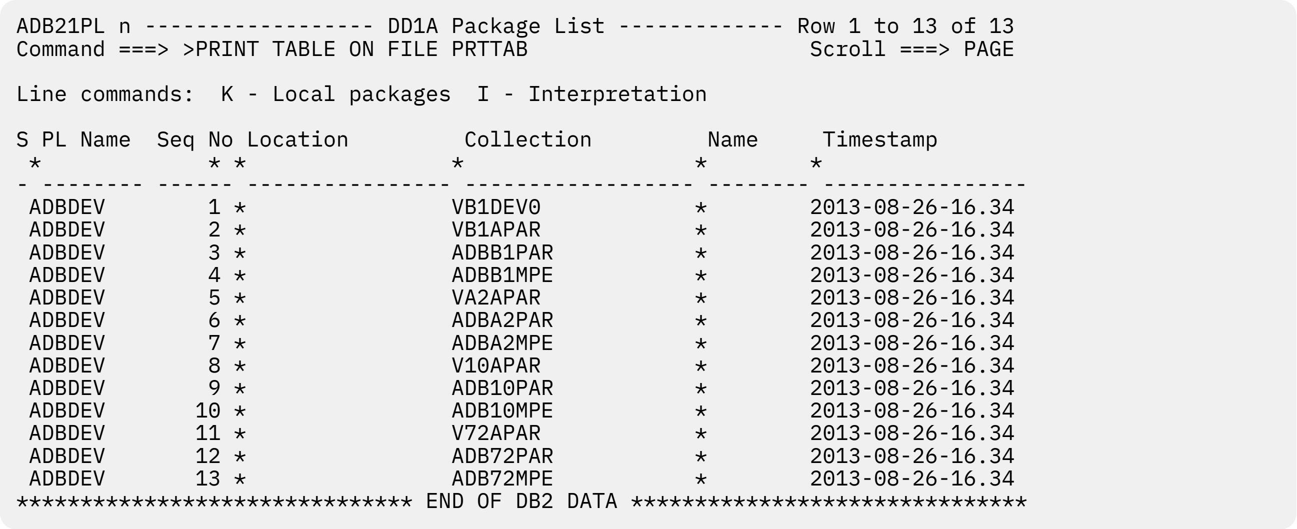


```
ADB21PL n ------------------ DD1A Package List ------------- Row 1 to 13 of 13
Command ===>                                                  Scroll ===> PAGE

Line commands:  K - Local packages  I - Interpretation

S PL Name  Seq No Location         Collection         Name     Timestamp
 *             * *                *                  *        *
- -------- ------ ---------------- ----------------- -------- ----------------
 ADBDEV        1 *                VB1DEV0            *       2013-08-26-16.34
 ADBDEV        2 *                VB1APAR            *       2013-08-26-16.34
 ADBDEV        3 *                ADBB1PAR           *       2013-08-26-16.34
 ADBDEV        4 *                ADBB1MPE           *       2013-08-26-16.34
 ADBDEV        5 *                VA2APAR            *       2013-08-26-16.34
 ADBDEV        6 *                ADBA2PAR           *       2013-08-26-16.34
 ADBDEV        7 *                ADBA2MPE           *       2013-08-26-16.34
 ADBDEV        8 *                V10APAR            *       2013-08-26-16.34
 ADBDEV        9 *                ADB10PAR           *       2013-08-26-16.34
 ADBDEV       10 *                ADB10MPE           *       2013-08-26-16.34
 ADBDEV       11 *                V72APAR            *       2013-08-26-16.34
 ADBDEV       12 *                ADB72PAR           *       2013-08-26-16.34
 ADBDEV       13 *                ADB72MPE           *       2013-08-26-16.34
******************************* END OF DB2 DATA *******************************
```

Step 4: Issue the command PRINT TABLE ON FILE

In the panel that contains the object that you want to print, issue the print command `>PRINT TABLE ON FILE PRINT` or `>PRINT TABLE ON FILE PRTTAB`, depending on the DD name that you specified in Step 2.

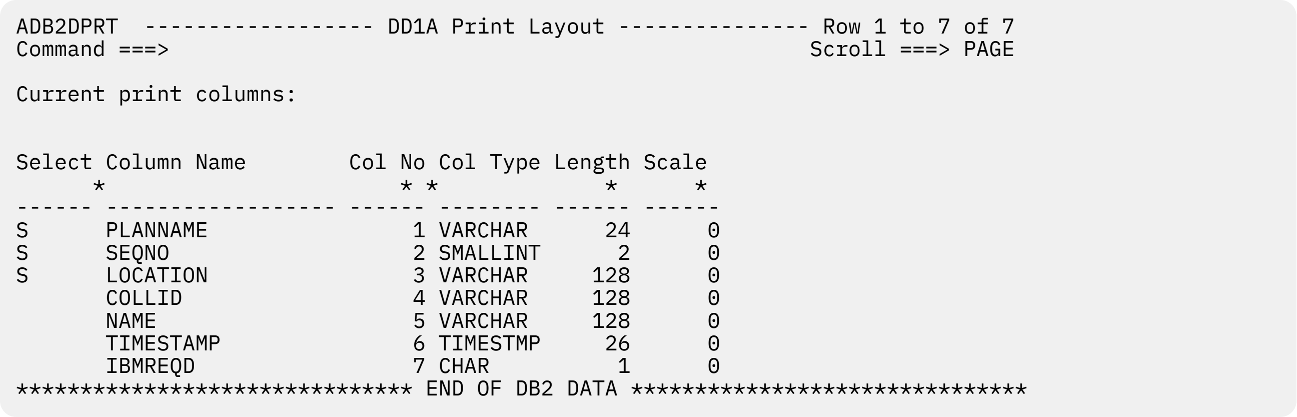
The ISPF command prefix (>) ensures that the application-level command PRINT TABLE ON FILE overrides the standard ISPF PRINT command from the ISPF command table ISPCMDS. Alternatively, PRT TABLE ON FILE can be issued instead of >PRINT TABLE ON FILE.



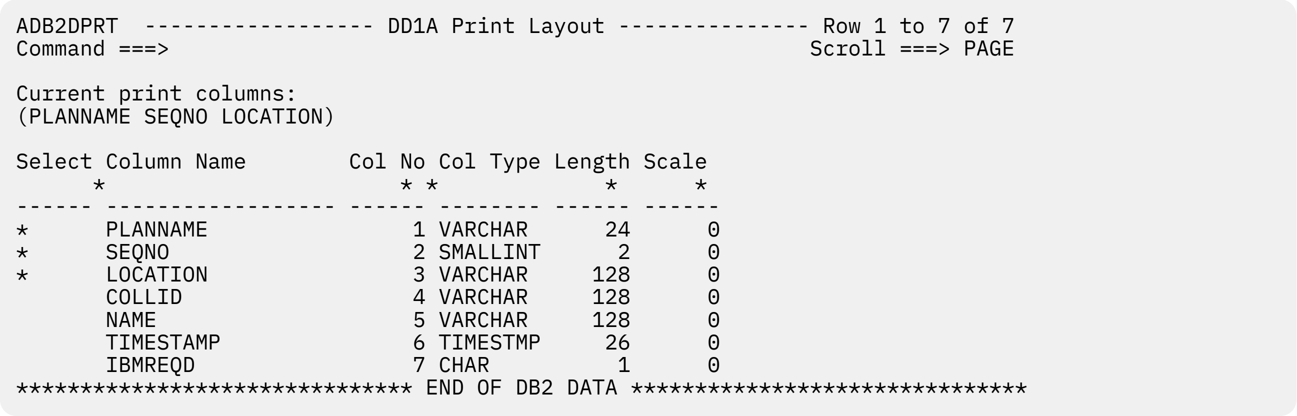
```
ADB21PL n ------------------ DD1A Package List ------------- Row 1 to 13 of 13
Command ===> >PRINT TABLE ON FILE PRTTAB                      Scroll ===> PAGE

Line commands:  K - Local packages  I - Interpretation

S PL Name  Seq No Location         Collection         Name     Timestamp
 *             * *                *                  *        *
- -------- ------ ---------------- ----------------- -------- ----------------
 ADBDEV        1 *                VB1DEV0            *       2013-08-26-16.34
 ADBDEV        2 *                VB1APAR            *       2013-08-26-16.34
 ADBDEV        3 *                ADBB1PAR           *       2013-08-26-16.34
 ADBDEV        4 *                ADBB1MPE           *       2013-08-26-16.34
 ADBDEV        5 *                VA2APAR            *       2013-08-26-16.34
 ADBDEV        6 *                ADBA2PAR           *       2013-08-26-16.34
 ADBDEV        7 *                ADBA2MPE           *       2013-08-26-16.34
 ADBDEV        8 *                V10APAR            *       2013-08-26-16.34
 ADBDEV        9 *                ADB10PAR           *       2013-08-26-16.34
 ADBDEV       10 *                ADB10MPE           *       2013-08-26-16.34
 ADBDEV       11 *                V72APAR            *       2013-08-26-16.34
 ADBDEV       12 *                ADB72PAR           *       2013-08-26-16.34
 ADBDEV       13 *                ADB72MPE           *       2013-08-26-16.34
******************************* END OF DB2 DATA *******************************
```

Step 5: Select the content that you want to print and exit

In the Print Layout (ADB2DPRT) panel, you can select the columns of data that you want to print:



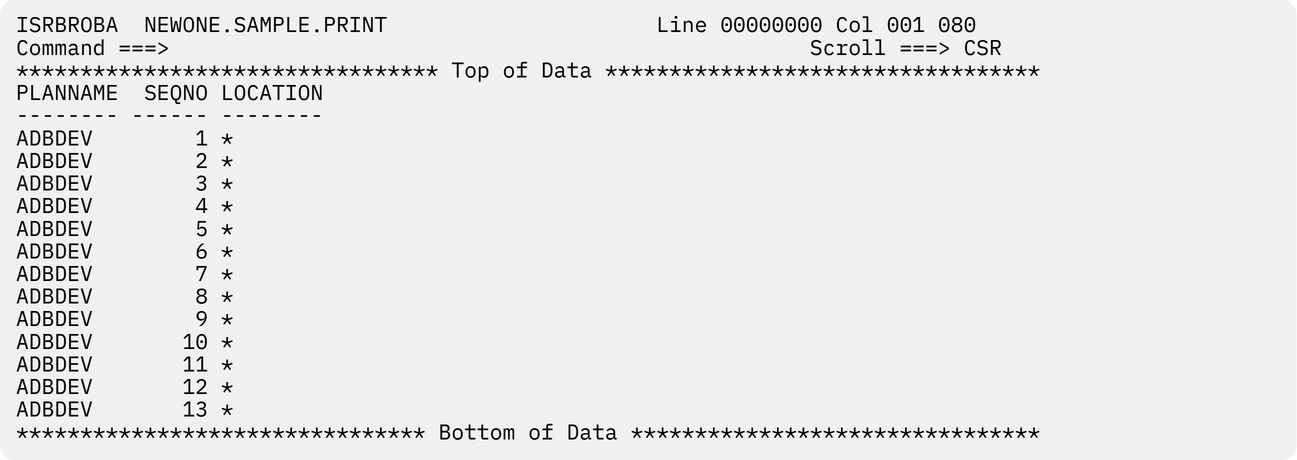
```
ADB2DPRT  ------------------ DD1A Print Layout --------------- Row 1 to 7 of 7
Command ===>                                                  Scroll ===> PAGE

Current print columns:


Select Column Name        Col No Col Type Length Scale
      *                       * *             *      *
------ ------------------ ------ -------- ------ ------
S      PLANNAME                1 VARCHAR      24      0
S      SEQNO                   2 SMALLINT      2      0
S      LOCATION                3 VARCHAR     128      0
       COLLID                  4 VARCHAR     128      0
       NAME                    5 VARCHAR     128      0
       TIMESTAMP               6 TIMESTMP     26      0
       IBMREQD                 7 CHAR          1      0
******************************* END OF DB2 DATA *******************************
```



```
ADB2DPRT  ------------------ DD1A Print Layout --------------- Row 1 to 7 of 7
Command ===>                                                  Scroll ===> PAGE

Current print columns:
(PLANNAME SEQNO LOCATION)

Select Column Name        Col No Col Type Length Scale
      *                       * *             *      *
------ ------------------ ------ -------- ------ ------
*      PLANNAME                1 VARCHAR      24      0
*      SEQNO                   2 SMALLINT      2      0
*      LOCATION                3 VARCHAR     128      0
       COLLID                  4 VARCHAR     128      0
       NAME                    5 VARCHAR     128      0
       TIMESTAMP               6 TIMESTMP     26      0
       IBMREQD                 7 CHAR          1      0
******************************* END OF DB2 DATA *******************************
```

Result: View the data set

In the standard Browse data panel (ISRBROBA) in z/OS ISPF, you can view the data set.



```
ISRBROBA  NEWONE.SAMPLE.PRINT                     Line 00000000 Col 001 080
Command ===>                                                  Scroll ===> CSR
********************************* Top of Data **********************************
PLANNAME  SEQNO LOCATION
-------- ------ --------
ADBDEV       1 *
ADBDEV       2 *
ADBDEV       3 *
ADBDEV       4 *
ADBDEV       5 *
ADBDEV       6 *
ADBDEV       7 *
ADBDEV       8 *
ADBDEV       9 *
ADBDEV      10 *
ADBDEV      11 *
ADBDEV      12 *
ADBDEV      13 *
******************************** Bottom of Data ********************************
```

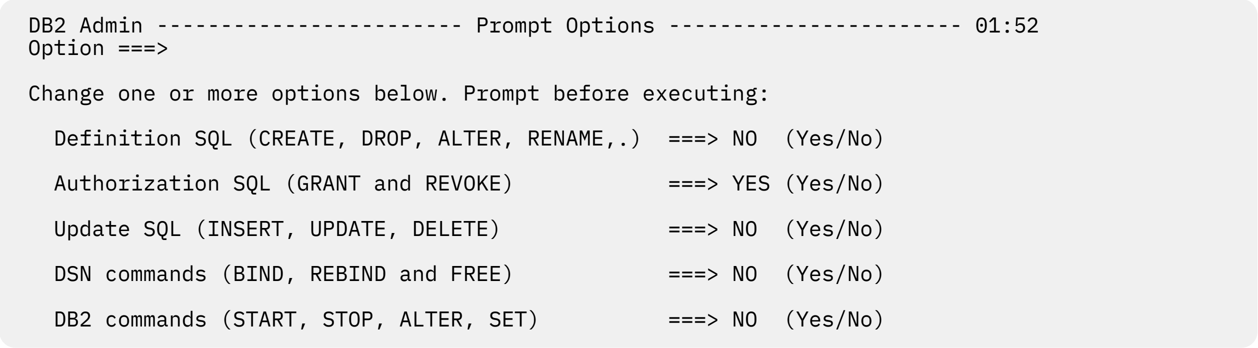
Changing Db2 Admin Tool prompt options

You can specify that you want Db2 Admin Tool to prompt you before running certain statements and commands. That prompt asks you to choose whether you want to run the statement or command immediately, edit it first, run it in a batch job, or add it to a work statement list (WSL). If you do not specify that you want to be prompted, the statement or command is run immediately.

Procedure

To change the prompt options:

1. Open the **DB2 Admin Options (ADB2P)** panel by performing one of the following actions:
   - On the **DB2 Administration Menu (ADB2)** panel, specify option P, and press Enter.
   - Issue the OPTIONS primary command on any Db2 Admin Tool panel.
2. On the **DB2 Admin Options (ADB2P)** panel, specify option PR, and press Enter.
3. On the **Prompt Options (ADB2PRMT)** panel, specify your preferences. To be prompted before the statement or command is run, specify Yes. Otherwise, specify No.



```
DB2 Admin ------------------------ Prompt Options ----------------------- 01:52
Option ===>

Change one or more options below. Prompt before executing:

  Definition SQL (CREATE, DROP, ALTER, RENAME,.)  ===> NO  (Yes/No)

  Authorization SQL (GRANT and REVOKE)            ===> YES (Yes/No)

  Update SQL (INSERT, UPDATE, DELETE)             ===> NO  (Yes/No)

  DSN commands (BIND, REBIND and FREE)            ===> NO  (Yes/No)

  DB2 commands (START, STOP, ALTER, SET)          ===> NO  (Yes/No)
```

*Figure 106. **Prompt Options (ADB2PRMT)** panel*

You can specify whether you want to be prompted for the following types of statements and commands:

Definition SQL (CREATE, DROP, ALTER, RENAME,.)
Any SQL statement that changes the definition of an object, such as CREATE, ALTER, DROP, and RENAME.

Authorization SQL (GRANT and REVOKE)
GRANT and REVOKE SQL statements.

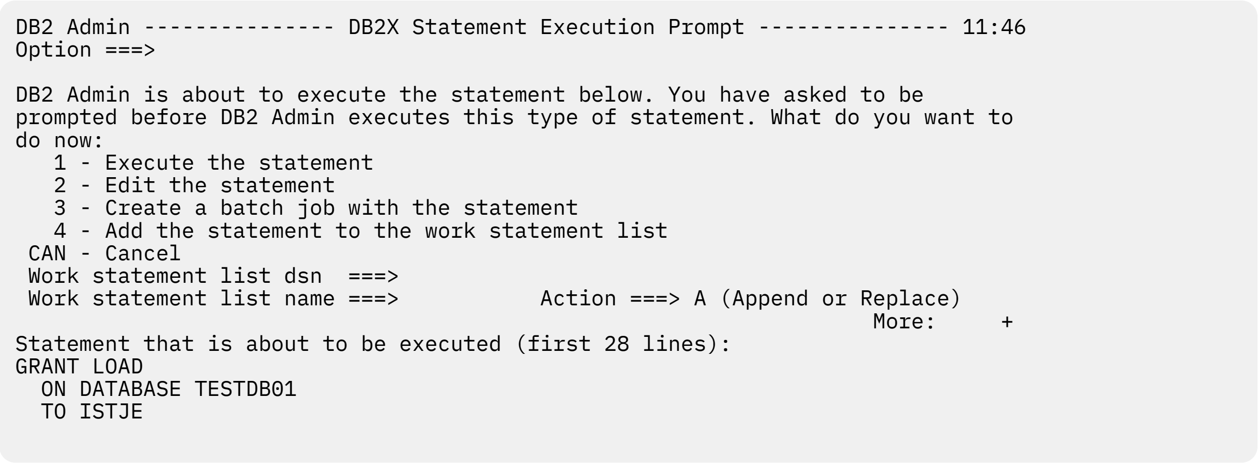
Update SQL (INSERT, UPDATE, DELETE)
INSERT, UPDATE, and DELETE statements.

DSN commands (BIND, REBIND and FREE)
DSN command statements such as those for BIND, REBIND, and FREE.

DB2 commands (START, STOP, ALTER, SET)
Db2 commands that change the state of an object or the system, such as START, STOP, ALTER, and SET.

When you later try to run the statement or command for which you requested a prompt, the **Statement Execution Prompt (ADB2PSTM)** panel is displayed.

For example, if you request a prompt before authorization statements are run [**Authorization SQL (GRANT and REVOKE** = YES], the **Statement Execution Prompt (ADB2PSTM)** panel is displayed when a request to grant load access to database TESTDB01 is made:



```
DB2 Admin --------------- DB2X Statement Execution Prompt --------------- 11:46
Option ===>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
   1 - Execute the statement
   2 - Edit the statement
   3 - Create a batch job with the statement
   4 - Add the statement to the work statement list
 CAN - Cancel
 Work statement list dsn  ===>
 Work statement list name ===>           Action ===> A (Append or Replace)
                                                                  More:     +
Statement that is about to be executed (first 28 lines):
GRANT LOAD
  ON DATABASE TESTDB01
  TO ISTJE
```

*Figure 107. **Statement Execution Prompt (ADB2PSTM)** panel – granting authorizations*

When more than one SQL statement is to be run, the following message is included on the **Statement Execution Prompt (ADB2PSTM)** panel:

```
(Add an A for all stmts. A is supported for options 1, 3, and 4. For example 1A – Execute
all statements)
```

In this case, the following additional options are available:

1A

Runs all statements.

Restriction: The “A” part of this option does not apply to Change Management. If you specify 1A and then subsequently specify that you want to use Change Management [on the **Change Management Prompt (ADB2CMPR)** panel], you must register each statement individually; you need to navigate through the **Change Management Prompt (ADB2CMPR)** panel and the **Register Change (ADB2CMRG)** panel for each statement. However, you can make each statement part of the same change by specifying the same change owner and name combination for each statement.

3A

Runs all statements in batch mode.

4A

Adds all statements to a work statement list. If **Action** is A (Append), the statements are added to the end of the work statement list. If **Action** is R (Replace), the work statement list is erased and then the statements are added.

Managing session scope for global variables

Create and manage global variables that you want to apply only to the current session.

About this task

Session scope variables override the default values of global variables.

The following conditions apply to session scope variables:

- They can be Db2 built-in global variables or user-defined global variables.
- They are active only for the current session and only when explicitly set.
- Their values may be used in the normal processing of other SQL statements such as DELETE, INSERT, SELECT, or UPDATE.

Procedure

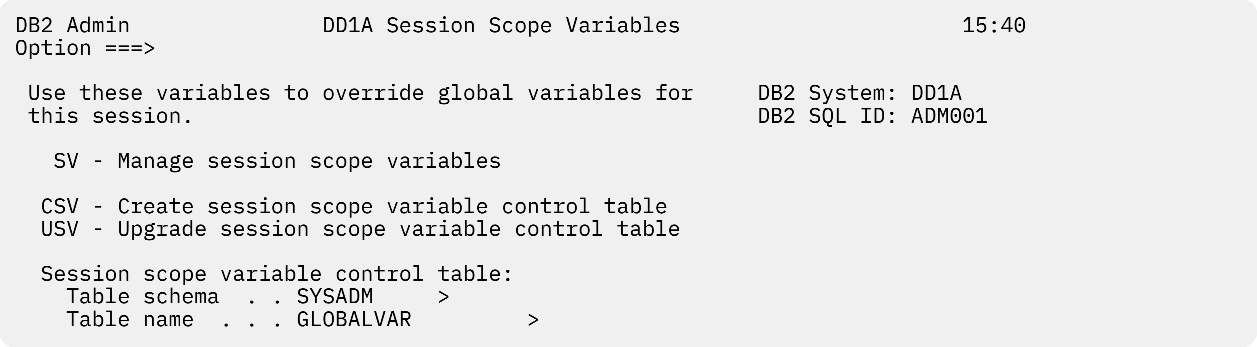
1. On the **DB2 Administration Menu (ADB2)** panel, specify option P, and press Enter.

Alternatively, issue the OPTIONS primary command on any Db2 Admin Tool panel.

The **DB2 Admin Options (ADB2P)** panel is displayed.

2. Specify option SV, and press Enter.

   The **Session Scope Variables (ADBPPSV)** panel is displayed, as shown in the following figure:

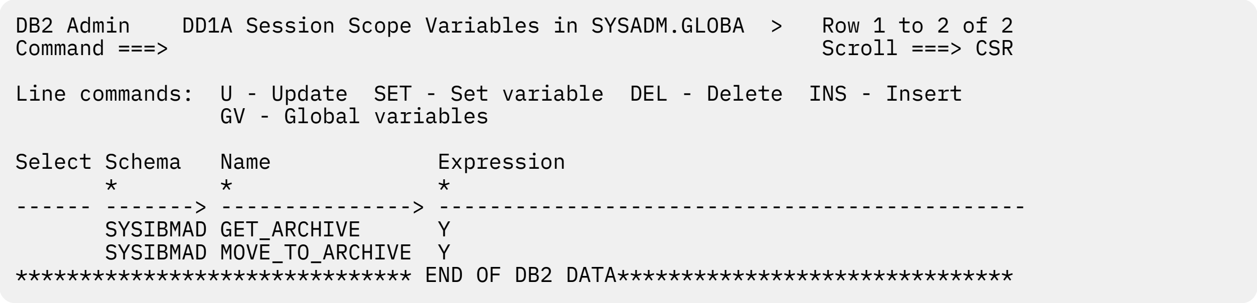


```
DB2 Admin               DD1A Session Scope Variables                      15:40
Option ===>

 Use these variables to override global variables for     DB2 System: DD1A
 this session.                                            DB2 SQL ID: ADM001

   SV - Manage session scope variables

  CSV - Create session scope variable control table
  USV - Upgrade session scope variable control table

  Session scope variable control table:
    Table schema  . . SYSADM     >
    Table name  . . . GLOBALVAR         >
```

*Figure 108. **Session Scope Variables (ADBPPSV)** panel*

3. Specify option SV, and press Enter.

   The **Session Scope Variables in schema.table (ADBPPSV1)** panel, as shown in the following figure.



```
DB2 Admin    DD1A Session Scope Variables in SYSADM.GLOBA  >   Row 1 to 2 of 2
Command ===>                                                   Scroll ===> CSR

Line commands:  U - Update  SET - Set variable  DEL - Delete  INS - Insert
                GV - Global variables

Select Schema   Name             Expression
       *        *                *
------ -------> ---------------> ----------------------------------------------
       SYSIBMAD GET_ARCHIVE      Y
       SYSIBMAD MOVE_TO_ARCHIVE  Y
******************************* END OF DB2 DATA******************************
```

*Figure 109. **Session Scope Variables in schema.table (ADBPPSV1)** panel*

4. Use line commands to manage the session variables.

Consolidating messages into a single file

Some batch interface jobs produce multiple output files, each of which contain a number of messages. To simplify troubleshooting, you can consolidate these messages into one file, or data set. You can do this consolidation for batch jobs for functions such as Change Management (CM) batch interface, ALT, Compare, and ADBTEP2.

About this task

When you request that messages are consolidated into one file, the messages are still included in the original output files. They are just repeated in the consolidated file. This consolidated file also includes any warning messages about unsupported DML statements and any requested object comparison reports.

Procedure

To consolidate messages into a single file, in the batch interface JCL procedure, add the ADBMSGS DD statement to specify the data set where you want the messages consolidated.

Example ADBMSGS DD statements:

-
  ```
  //ADBMSGS DD SYSOUT=*
  ```

-
  ```
  //ADBMSGS DD DISP=(MOD,CATLG),DSN=TS5781.SAMPLE.OUTPUT5,
  // SPACE=(CYL,(2,1))
  ```

-
  ```
  //ADBMSGS DD DISP=(MOD,CATLG),DSN=TS5781.SAMPLE.OUTPUT5,
  // SPACE=(CYL,(2,1)),
  // DCB=(LRECL=137,RECFM=VBA,DSORG=PS)
  ```

This data set that you defined to contain the consolidated messages is referred to as the *ADBMSGS data set*.

Tip: ADBMSGS always includes a list of the initial and profile values for CM batch parameters that are not blank. Additionally, if you want ADBMSGS to include the CM batch parameter values that were provided in an invocation override and the final CM batch parameter values that were used, set the CM batch parameter list_options to Y.

The following snippets show examples of content that is written to the ADBMSGS data set. Notice that messages in ADBMSGS include the module name, such as ADBCDTS or ADB2REE.

The following example shows how the Change Management (CM) batch parameter values are reported. Notice that in this example, three lists of parameter values are included, because the CM batch parameter list_options was set to Y. The beginning of each of these lists is identified by message ADB7957I.

```
1ADB7956I ADBCCM   CM batch: Start of CM batch

 ADB7957I ADBCCM   CM batch parameters-init and PROF:
                   ADBTEP2_RETRY_DEPRECATED_OBJ='YES';
                   AUTH_SWITCH_USERID='<NONE>';
                   AUTH_SWITCHING_ENABLED='Y';
                   JOB_JCLLIB_LINE_1='//ADBPROC JCLLIB ORDER=TSLHC.CMBATCH.PROCLIB';
                   JOB_PARM_LINE_1='S=RS22';
                   LIST_OPTIONS='N';
                   LOAD_ACCELERATED_TABLES_LOCK_MODE='TABLESET';
                   PLAN_NAME_IS_ALTERNATE='Y';
                   SCOPE_WARNING='NO';
                   SSID='DC1A';
                   USE_DSNUTIL_SP='YES';
                   UTIL_CLONE_TEMPLATE_COPYDDN1_USE='S';
                   UTIL_CLONE_TEMPLATE_COPYDDN2_USE='S';
                   UTIL_CLONE_TEMPLATE_DISCARDDN_USE='S';
                   UTIL_CLONE_TEMPLATE_ERRDDN_USE='S';
                   UTIL_CLONE_TEMPLATE_FCCOPYDDN_USE='S';
                   UTIL_CLONE_TEMPLATE_LOBCOL_USE='S';
                   UTIL_CLONE_TEMPLATE_MAPDDN_USE='S';
                   UTIL_CLONE_TEMPLATE_PUNCHDDN_USE='S';
                   UTIL_CLONE_TEMPLATE_RECOVERYDDN1_USE='S';
                   UTIL_CLONE_TEMPLATE_RECOVERYDDN2_USE='S';
                   UTIL_CLONE_TEMPLATE_UNLDDN_USE='S';
                   UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDN_USE='S';
                   UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDNC_USE='S';
                   UTIL_CLONE_TEMPLATE_UNLOAD_UNLDDN_USE='S';
                   UTIL_CLONE_TEMPLATE_UNLOAD_UNLDDNC_USE='S';
                   UTIL_CLONE_TEMPLATE_WORKDDN1_USE='S';
                   UTIL_CLONE_TEMPLATE_WORKDDN2_USE='S';
                   UTIL_CLONE_TEMPLATE_XMLCOL_USE='S';
                   UTIL_REORG_INDEX_SORTNUM='4';
                   UTIL_REORG_SORTNUM='4';
                   UTIL_TEMPLATE_COPYDDN1_USE='S';
                   UTIL_TEMPLATE_COPYDDN2_USE='S';
                   UTIL_TEMPLATE_DISCARDDN_USE='S';
                   UTIL_TEMPLATE_ERRDDN_USE='S';
                   UTIL_TEMPLATE_FCCOPYDDN_USE='S';
                   UTIL_TEMPLATE_LOBCOL_USE='S';
                   UTIL_TEMPLATE_MAPDDN_USE='S';
                   UTIL_TEMPLATE_PUNCHDDN_USE='S';
                   UTIL_TEMPLATE_RECOVERYDDN1_USE='S';
                   UTIL_TEMPLATE_RECOVERYDDN2_USE='S';
                   UTIL_TEMPLATE_UNLDDN_USE='S';
                   UTIL_TEMPLATE_UNLOAD_PUNCHDDN_USE='S';
                   UTIL_TEMPLATE_UNLOAD_PUNCHDDNC_USE='S';
                   UTIL_TEMPLATE_UNLOAD_UNLDDN_USE='S';
                   UTIL_TEMPLATE_UNLOAD_UNLDDNC_USE='S';
                   UTIL_TEMPLATE_WORKDDN1_USE='S';
                   UTIL_TEMPLATE_WORKDDN2_USE='S';
                   UTIL_TEMPLATE_XMLCOL_USE='S';
 ADB7957I ADBCCM   CM batch parameters-PARMS DD applied:
                   ACCEPT_FL='507';
                   ACTION_ANALYZE_CHANGE='Y';
                   ACTION_BUILD_RUN_JOB='Y';
                   ACTION_COMPARE='N';
```

```
                   ACTION_IMPORT_CHANGE='Y';
                   ACTION_RUN_CHANGE='Y';
                   ADBTEP2_RETRY_DEPRECATED_OBJ='YES';
                   AUTH_SWITCH_USERID='<NONE>';
                   AUTH_SWITCHING_ENABLED='Y';
                   CHANGE_NAME='ADB10376';
                   EXISTING_DATA_SET_ACTION='REPLACE';
                   IMPORT_PENDING_CHANGE_ACTION='S';
                   JOB_JCLLIB_LINE_1='//ADBPROC JCLLIB ORDER=TSLHC.CMBATCH.PROCLIB';
                   JOB_PARM_LINE_1='S=RS22';
                   LIST_OPTIONS='Y';
...
 ADB7957I ADBCCM   CM batch parameters-final:
                   ACCEPT_FL='507';
                   ACTION_ANALYZE_CHANGE='Y';
                   ACTION_BUILD_RUN_JOB='Y';
                   ACTION_CANCEL_CHANGE='N';
                   ACTION_COMPARE='N';
                   ...
                   LIST_OPTIONS='Y';
...
 ADB7956I ADBCCM   CM batch: Start of Import phase
1ADB7953I ADB2CID  ADB2CID - Import changes - 2020-07-06 19:05
 ADB7953I ADBCDTS  ADBCDTS - Create Version File from DDL File
2020-07-06 19:05
 ADB7953I ADBCDTS  Using DB2 DECP Version 1215  startup parameters for SSID DC1A
...
```

The following example shows additional content that can be included in ADMSGS:

```
1ADB7956I ADBCCM   CM batch: Start of CM batch
...
 ADB7953I ADBCDTS  DB2 DSNHDECP values for this run :
 ADB7953I ADBCDTS  DB2 Version, Release and Mod Level : 1215            Default CCSID for EBCDIC
SBCS      : 00037
 ADB7953I ADBCDTS  Decimal point option               : '.'             Default CCSID for EBCDIC
Mixed     : 00002
 ADB7953I ADBCDTS  Subsystem ID                       : DC1A            Default CCSID for EBCDIC
DBCS      : 00002
 ADB7953I ADBCDTS  Graphic for DBCS data              : No              Default CCSID for ASCII
SBCS       : 00437
 ADB7953I ADBCDTS  Date format                        : USA             Default CCSID for ASCII
Mixed      : 00002
 ADB7953I ADBCDTS  Time format                        : ISO             Default CCSID for ASCII
DBCS       : 00002
 ADB7953I ADBCDTS  Default encoding scheme            : EBCDIC          Default CCSID for UNICODE
SBCS     : 00367
 ADB7953I ADBCDTS  DB2 Version 12 New Function Mode   : No              Default CCSID for UNICODE
Mixed    : 01208
 ADB7953I ADBCDTS                                                       Default CCSID for UNICODE
DBCS     : 01200
 ADB7713I ADBCDTS  The DDL reader is processing under the authorization ID for the TSLHC schema. The
authorization
                   ID can be changed by the SET CURRENT SCHEMA statement.

 ...

 ADB1057I ADB2REE  Copy Stogroup Grant(s)  : No
 ADB1058I ADB2REE  Copy Database Grant(s)  : Yes    Copy Table Space Grant(s): Yes    Copy Table
Grant(s)  . . : Yes
 ADB1059I ADB2REE  Copy View Grant(s)  . . : Yes    Copy authorizations on referenced
schema(s)  . . . . . . . : Yes
 ADB1060I ADB2REE  Copy U.def type Grant(s): Yes    Copy Function Grant(s) . : Yes    Copy Procedure
Grant(s)  : Yes
 ADB1198I ADB2REE  Copy Sequence Grant(s)  : Yes    Copy Variable Grant(s) . : Yes
 ADB1079I ADB2REE  Insert COMMIT statement after every definition.
 ADB1085I ADB2REE  RE will generate all parameters even if they take default values.
 ADB1086I ADB2REE  RE will generate all implicitly referenced objects.
 ADB1095I ADB2REE  RE will convert auxiliary table requests to the respective base table.
 ADB1096I ADB2REE  DB2 pending changes will be merged into CREATE statements.
 ADB1101I ADB2REE  XML type modifiers defined using the following syntax
                   URI target-namespace LOCATION schema-location-uri
                   will be generated using the following syntax
                   ID registered-XML-schema-name
                   because the DB2 catalog does not have enough externalized information to distinguish
between
                   these two types of syntax.
 ADB1609W ADB2REM  View DB2IVP0.VLE_FSA_REPORTING_BP not found
 ADB1609W ADB2REM  Alias DB2IVP0.VLE_FSA_REPORTING_BP not found
 ADB1609W ADB2REM  Table DB2IVP0.VLE_FSA_REPORTING_BP not found
 ADB1609W ADB2REP  Synonym DB2IVP0.VLE_FSA_REPORTING_BP not found
 ADB1176I ADB2REE  ADB2GEN - Summary of catalog records written
                   Number of catalog records written:       1
 ADB1024I ADB2REE  ADB2GEN - Ended with warnings
1ADB7956I ADBCCMM  ADBCCMM - Merge Version Files
 ADB8998I ADB2CMP  ==================  Start of Report
============================================================

           Parameters for this run :

           ANALYZE mode produces change statements intended for the Analyze process of CM,
           report has content (messages and actions) that are unique to this mode.

           Suppress DROP of objects  : No
           Suppress DROP of columns  : No
           Suppress adding columns   : No


 ANALYZE OBJECT COMPARISON REPORT
 ================================

 Original values will be applied to ignored fields of new objects.

           Source:
                   Extracted from location                  at 2019-04-19 14:41 by TSLHC
           Target:
                   Extracted from location                  at 2019-04-19 14:41 by TSLHC

           Target system is DB2 Release 1215

 View DB2IVP0.VLE_FSA_REPORTING_BP not found on target
   New View DB2IVP0.VLE_FSA_REPORTING_BP will be added

 GOC2CMP - Ended normally in ANALYZE mode

 ADB8999I ADB2CMP  ==================  End of Report
============================================================

1ADB9354I ADBCUPC  Analyze completed successfully.  The change status has been updated to
                   ANALYZED for change "TSLHC"."CM1_ADB2699" (ID= 2).
 ...
 ADB7956I ADBCCM   CM batch: End of CM batch
```

Db2 Admin Tool data type conversions

Db2 Admin Tool supports conversions between certain data types.

The following tables show the supported conversions. In the first table, X indicates that Db2 Admin Tool supports the data type conversion.

Table 13. Db2 Admin Tool data type conversions, part 1

	New data type:												
Original data type:	**sm. int.**	**int.**	**float**	**dec.**	**char.**	**vchar.**	**long vchar.**	**graph**	**var. graph.**	**long vgr.**	**date**	**time**	**time st.**
small integer	X	X	X	X[1]	X	X							
integer	X[1]	X	X	X[1]	X	X							
float	X	X	X	X									
decimal	X[1]	X[1]	X	X[1]	X	X					X	X	X
character	X	X		X	X[1]	X[1]	X				X[2]	X[3]	X[4]
varchar	X	X		X	X[1]	X[1]	X				X[2]	X[3]	X[4]
long varchar					X[1]	X[1]	X				X[2]	X[3]	X[4]
graphic								X	X	X			
vgraphic								X	X	X			

Table 13. Db2 Admin Tool data type conversions, part 1 (continued)

	New data type:												
Original data type:	**sm. int.**	**int.**	**float**	**dec.**	**char.**	**vchar.**	**long vchar .**	**graph**	**var. graph .**	**long vgr.**	**date**	**time**	**time st.**
long vgraphic								X	X	X			
date					X	X					X		X
time					X	X						X	
time stamp					X	X					X	X	X

Notes:

1. This conversion checks for truncation and number overflows. It is displayed during the ALT process and before job submission.
2. For this conversion to date format, the following load formats are valid:
   - dd.mm.yyyy
   - mm/dd/yyyy
   - yyyy-mm-dd
3. For this conversion to time format, the following load formats are valid:
   - hh.mm.ss
   - hh:mm AM
   - hh:mm PM
   - hh:mm:ss
4. For this conversion to time stamp format, the following load formats are valid:
   - yyyy-mm-dd-hh.mm.ss
   - yyyy-mm-dd-hh.mm.ss.nnnnnn

In the following table, an A or D indicates that Db2 Admin Tool supports the data type conversion. A indicates that the object action is ALTER. D indicates that the object action is DROP or DROP-SC.

Table 14. Db2 Admin Tool data type conversions, part 2

	New data type:											
Original data type:	**sm int**	**int**	**float**	**dec**	**char**	**vchar**	**long vchar**	**big int**	**dec float (16)**	**dec float (34)**	**binary**	**varbinary**
small integer								A	A	A		
integer								A	A	A		
float								D	A	A		
decimal								A[2]	A	A		
character											A[1]	A[1]
varchar											A[1]	A[1]
long varchar						A						
big integer	D	D	D	A					D	A		
dec float (16)	D	D	D	D				D		A		
dec float (34)	D	D	D	D				D	D			
binary											A	A
varbinary											A	A

Notes:

1. The original column must be defined as FOR BIT DATA.

2. Due to a potential issue when converting from DECIMAL(19,0) to BIGINT using a Db2 ALTER statement, Db2 Admin Tool performs a DROP instead along with data conversion to detect the data issue. For details, see ALTER TABLE (Db2 12 for z/OS).

If the truncation action chosen on panel ADB27CT is “Z” or “T”, the action is a DROP. Otherwise, the action is a DROP-SC. Changing NULL to NOT NULL requires a DROP operation.

Db2 Admin Tool with a large number of objects

Enterprise Resource Planning (ERP) applications are increasingly using Db2 for z/OS.

These ERP systems typically have a large number of objects, such as 1 000 databases, 10 000 to 30 000 table spaces, and 20 000 to 100 000 tables that have one or more indexes. Administering such large Db2 systems is a challenge, and when you use certain Db2 Admin Tool functions, you must take into account the large number of objects. In addition, the data sets that are allocated for Db2 Admin Tool and ISPF functions must be large enough to accommodate the large number of objects.

Topics:

- “ISPF work data sets” on page 251
- “Output data sets for GEN DDL ” on page 252
- “Other recommendations for a large number of objects” on page 252

ISPF work data sets

Db2 Admin Tool uses ISPF file tailoring services when generating batch jobs. The ISPF services uses preallocated work data sets when generating the JCL for the batch jobs. However, when you generate JCL for many objects, the preallocated ISPF work data sets might not be large enough.

The ISPF work data sets are either allocated by the TSO logon procedure or dynamically allocated based on ISPF customization parameters. When you generate batch jobs for many objects, you might need to have the allocations changed for the data sets with these ISPF DD names:

- ISPCTL*x*: points to the ISPF temporary data set default name SPFTEMP*x*.CNTL
- ISPWRK*x*: points to the ISPF temporary data set default name SPFTEMP*x*.WORK

Where *x* represents an ISPF logical screen name

Example: *x* = value 1-9, A-W

The recommended space allocation for these data sets is SPACE=(CYL,(1,5). This space allocation allows for generating batch jobs with 115,000 lines of JCL, using three extents. If you are experiencing space problems (x37 abends), contact your storage administrator to have the space allocations changed for the DD names listed. For additional information on ISPF temporary data sets, see Preallocate ISPF temporary data sets to VIO (z/OS 3.1.0).

If you have a RUN CM ABEND*x*37 failure related to the ISPCTL*x* or ISPWRK*x* DDs, you can resolve it in one of the following ways:

- Online: Use ANALYSE to generate RUN WSL
- Batch: Change the SADBSLIB skeleton member ADB2SPFB by modifying it for the default allocation for ISPWRK1 and ISPWRK2, as follows:

```
//ISPWRK1  DD DSN=&&ISPWRK1,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=256,BLKSIZE=256,DSORG=PS),
//            SPACE=(CYL,(5,10)),UNIT=SYSALLDA
//ISPWRK2  DD DSN=&&ISPWRK2,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=256,BLKSIZE=256,DSORG=PS),
//            SPACE=(CYL,(5,10)),UNIT=SYSALLDA
```

Output data sets for GEN DDL

When you use the Db2 Admin Tool GEN function to generate DDL for objects in the Db2 catalog, you can choose to place the DDL in different types of output data sets.

Specifically, you can choose to place the DDL in:

- An existing or new data set
- An existing or new work statement list (WSL) data set

When you generate DDL for a large number of objects and specify that a new data set be used, either a regular data set or a WSL data set, the default space allocation that Db2 Admin Tool uses might not be sufficient.

If you are experiencing x37 abends on the output data set (either regular or WSL) for the generated DDL, use a preallocated data set instead of a new data set. Define the DDL output data set with the following attributes:

```
RECFM=FB
LRECL=80
```

The generated DDL for all the objects in an ERP system can get very large, for example, 3 million lines of statements. The GEN DDL output data set for that number of statements would require 287 cylinders. You can use ISPF option 3.2 to preallocate a large data set. A WSL data set must be a partitioned data set.

Other recommendations for a large number of objects

You should follow certain recommendations when you use Db2 Admin Tool in an environment that has a large number of objects.

The following recommendations will help you use Db2 Admin Tool with a large number of objects:

- Reduce the number of objects for primary commands. Running Db2 Admin Tool primary commands on a very large number of objects can take some time and locks your ISPF session while the objects are being processed. If possible, when searching for objects in the Db2catalog (Db2 Admin Tool option 1), limit the number of objects by specifying a narrower search criteria.
- When searching for objects in the Db2 catalog (Db2 Admin Tool option 1), use a search criteria that allows Db2 to use indexes to retrieve the information that you need. For more information, see the online help for the **System Catalog** panel (ADB21).
- Add the recommended indexes to the Db2 catalog.
- Run RUNSTATS on the Db2 catalog.
- Ensure that there is free space on the DASD volumes that you are using. Db2 Admin Tool functions might need to expand the data sets beyond the primary allocation. Extending the data sets with secondary extents requires that the DASD volume has sufficient free space. If you are experiencing problems with space on data sets that have not reached their maximum extents, contact your storage administrator. The storage administrator might need to change the storage policy for these data sets to avoid the problems.
- Ensure that your batch jobs can get sufficient virtual storage. Some Db2 Admin Tool functions keep information in storage while processing through the objects. If you are experiencing out-of-storage abends, specify a large region size on the job card, for example, 64 MB. If you still experience abends, contact your system administrator because the installation limits in the system that you are using might be causing the problem.
- Ensure that your batch jobs can get sufficient CPU time. When you generate the DDL for a large number of objects, you might, depending on your installation settings and processor speed, need to add a TIME=*n* option on your job card. The recommended initial value for *n* is 180 (CPU minutes).

Tutorial

The topics in this information demonstrate how to navigate Db2 Admin Tool and introduce you to some of its major functions.

Comprehensive information about all of Db2 Admin Tool functionality is contained in Chapter 4, “Db2 management,” on page 267.

Remember: This tutorial is based on the sample database that is provided with Db2. If you do not have the sample database installed on your system, you can still follow along with the tutorial by using one of your own databases.

Figure 110 on page 253 shows the **DB2 Administration Menu (ADB2)** panel that is displayed when you start Db2 Admin Tool. The top of the panel shows the Db2 Admin Tool functions you can choose. The release level and mode of your Db2 subsystem affect the options, within the functions, that are available to you. The bottom of the panel shows other Db2 tools (in this case, Db2 Interactive and Db2 Object Comparison Tool) that can be invoked from the main menu; this is a customization option.

Running queries

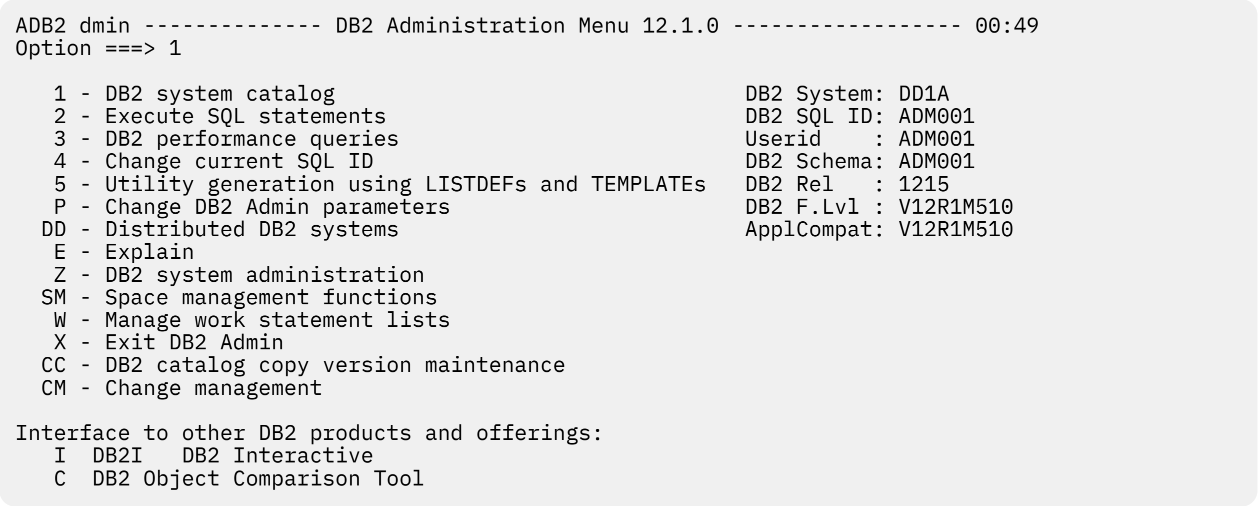
You can run queries to display and filter information about database objects.

About this task

In this stage of the tutorial, you will view databases in the system catalog.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, in the **Option** field, specify 1, as shown in the following example:



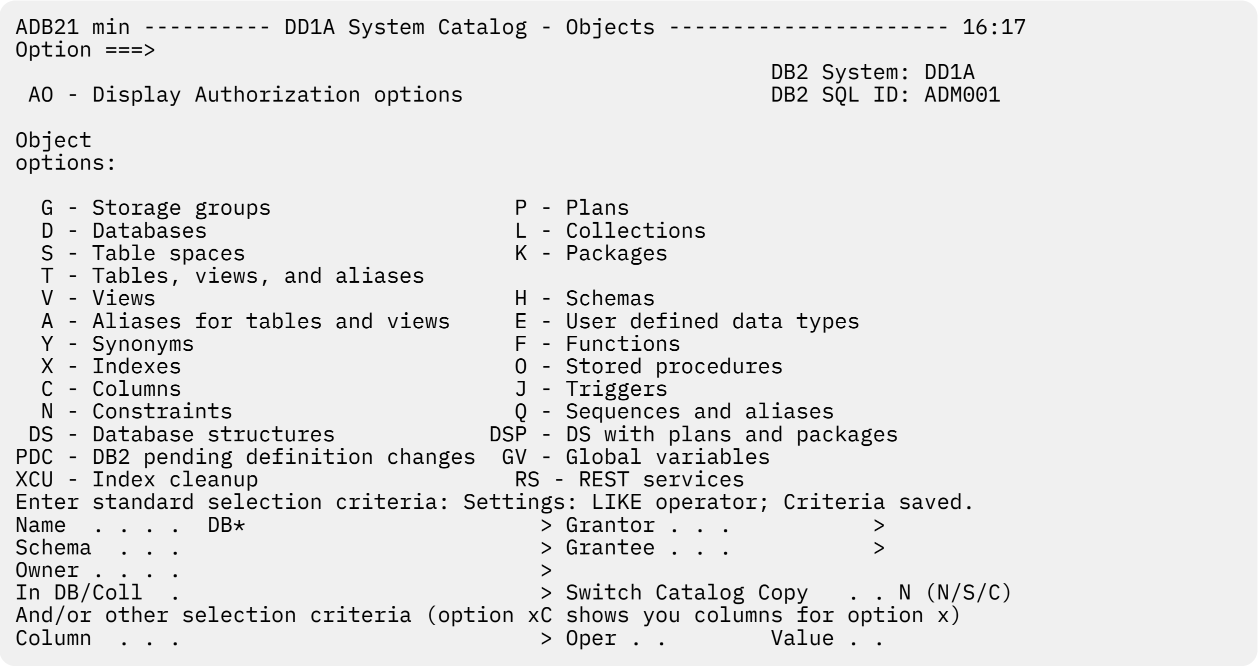
```
ADB2 dmin -------------- DB2 Administration Menu 12.1.0 ------------------ 00:49
Option ===> 1

   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   : 1215
   P - Change DB2 Admin parameters                      DB2 F.Lvl : V12R1M510
  DD - Distributed DB2 systems                          ApplCompat: V12R1M510
   E - Explain
   Z - DB2 system administration
  SM - Space management functions
   W - Manage work statement lists
   X - Exit DB2 Admin
  CC - DB2 catalog copy version maintenance
  CM - Change management

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

*Figure 110. **DB2 Administration Menu (ADB2)** panel*

2. Press Enter.

   The **System Catalog (ADB21)** panel is displayed.

3. Specify option D for databases, as shown in the following figure:

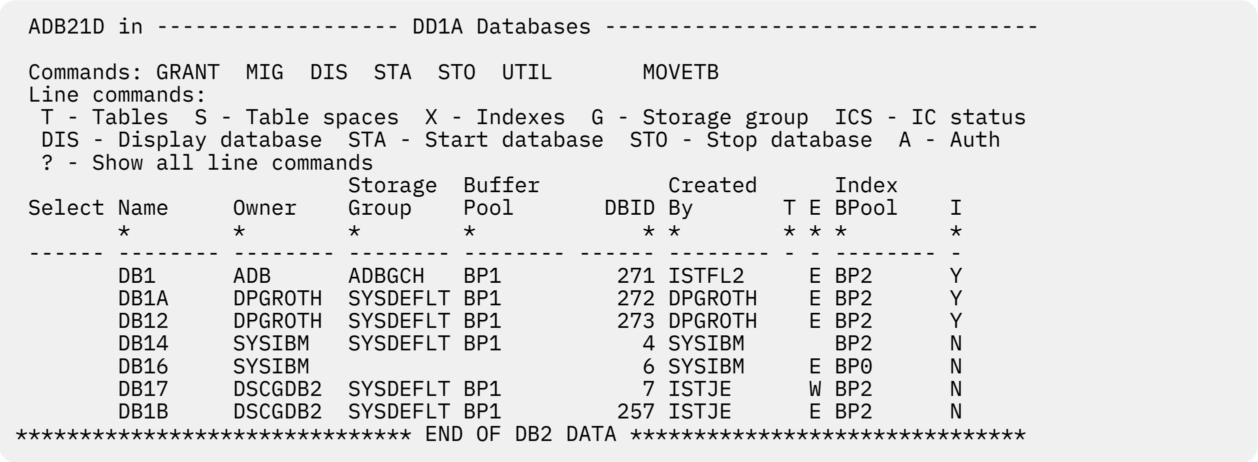


```
ADB21 min ---------- DD1A System Catalog - Objects --------------------- 16:17
Option ===>
                                                           DB2 System: DD1A
 AO - Display Authorization options                        DB2 SQL ID: ADM001

Object
options:

  G - Storage groups                   P - Plans
  D - Databases                        L - Collections
  S - Table spaces                     K - Packages
  T - Tables, views, and aliases
  V - Views                            H - Schemas
  A - Aliases for tables and views     E - User defined data types
  Y - Synonyms                         F - Functions
  X - Indexes                          O - Stored procedures
  C - Columns                          J - Triggers
  N - Constraints                      Q - Sequences and aliases
 DS - Database structures            DSP - DS with plans and packages
PDC - DB2 pending definition changes  GV - Global variables
XCU - Index cleanup                    RS - REST services
Enter standard selection criteria: Settings: LIKE operator; Criteria saved.
Name  . . . .  DB*                       > Grantor . . .           >
Schema  . . .                            > Grantee . . .           >
Owner . . . .                            >
In DB/Coll  .                            > Switch Catalog Copy   . . N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .                            > Oper . .        Value . .
```

*Figure 111. **System Catalog (ADB21)** panel - object options*

4. Press Enter.

   The **Databases (ADB21D)** panel is displayed, as shown in the following figure. This panel lists all databases that meet the search criteria.



```
 ADB21D in ------------------- DD1A Databases ----------------------------------

 Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
 Line commands:
  T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
  DIS - Display database  STA - Start database  STO - Stop database  A - Auth
  ? - Show all line commands
                          Storage  Buffer          Created      Index
 Select Name     Owner    Group    Pool       DBID By       T E BPool    I
        *        *        *        *            * *         * * *        *
 ------ -------- -------- -------- -------- ------ -------- - - -------- -
        DB1      ADB      ADBGCH   BP1         271 ISTFL2     E BP2      Y
        DB1A     DPGROTH  SYSDEFLT BP1         272 DPGROTH    E BP2      Y
        DB12     DPGROTH  SYSDEFLT BP1         273 DPGROTH    E BP2      Y
        DB14     SYSIBM   SYSDEFLT BP1           4 SYSIBM       BP2      N
        DB16     SYSIBM                          6 SYSIBM     E BP0      N
        DB17     DSCGDB2  SYSDEFLT BP1           7 ISTJE      W BP2      N
        DB1B     DSCGDB2  SYSDEFLT BP1         257 ISTJE      E BP2      N
****************************** END OF DB2 DATA ******************************
```

Figure 112. Databases (ADB21D)

   You can filter the databases that are displayed by specifying a search argument in the **Name** field. You can use a percent sign (%) or an asterisk (*) as a wildcard character in your search argument.

What to do next

Press PF3 to return to the **System Catalog (ADB21)** panel to complete the next stage of the tutorial.

Running utilities

You can run Db2 utilities from Db2 Admin Tool.

Before you begin

Open the **System Catalog (ADB21)** panel. (On the **DB2 Administration Menu (ADB2)** panel, select option 1.)

About this task

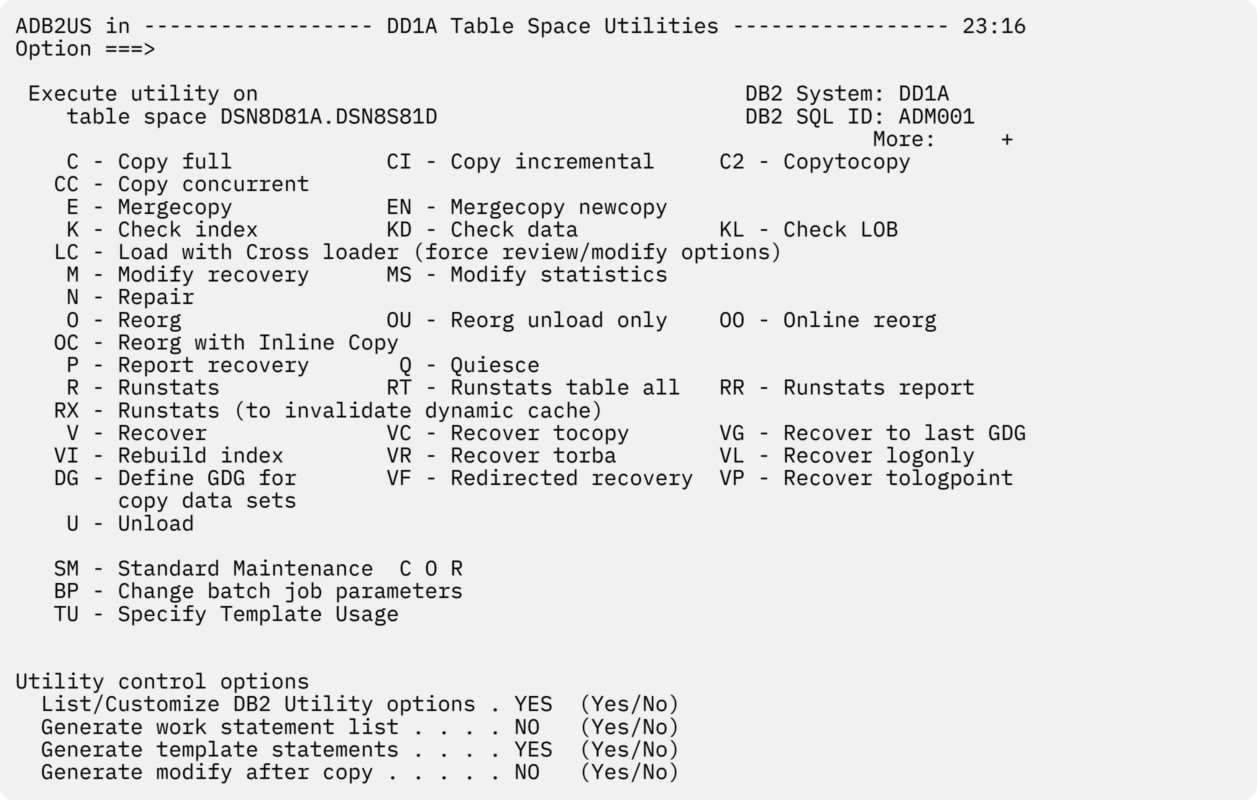
This tutorial shows you how to do the following tasks:

- Run utilities against a table space.
- Run the copy utility.
- View the tables in a table space.
- View the columns in a table.
- View the indexes in a table.

Procedure

1. On the **System Catalog (ADB21)** panel, specify option S, and press Enter.
2. On the **Table Spaces (ADB21S)** panel, specify the UTIL or UTL line command against the table space DSN8S81D, and press Enter.

   The **Table Space Utilities (ADB2US)** panel shows the utilities that can be run against the table space:



```
ADB2US in ------------------ DD1A Table Space Utilities ----------------- 23:16
Option ===>

 Execute utility on                                      DB2 System: DD1A
    table space DSN8D81A.DSN8S81D                        DB2 SQL ID: ADM001
                                                                   More:     +
    C - Copy full            CI - Copy incremental     C2 - Copytocopy
   CC - Copy concurrent
    E - Mergecopy            EN - Mergecopy newcopy
    K - Check index          KD - Check data           KL - Check LOB
   LC - Load with Cross loader (force review/modify options)
    M - Modify recovery      MS - Modify statistics
    N - Repair
    O - Reorg                OU - Reorg unload only    OO - Online reorg
   OC - Reorg with Inline Copy
    P - Report recovery       Q - Quiesce
    R - Runstats             RT - Runstats table all   RR - Runstats report
   RX - Runstats (to invalidate dynamic cache)
    V - Recover              VC - Recover tocopy       VG - Recover to last GDG
   VI - Rebuild index        VR - Recover torba        VL - Recover logonly
   DG - Define GDG for       VF - Redirected recovery  VP - Recover tologpoint
        copy data sets
    U - Unload

   SM - Standard Maintenance  C O R
   BP - Change batch job parameters
   TU - Specify Template Usage


Utility control options
  List/Customize DB2 Utility options . YES  (Yes/No)
  Generate work statement list . . . . NO   (Yes/No)
  Generate template statements . . . . YES  (Yes/No)
  Generate modify after copy . . . . . NO   (Yes/No)
```

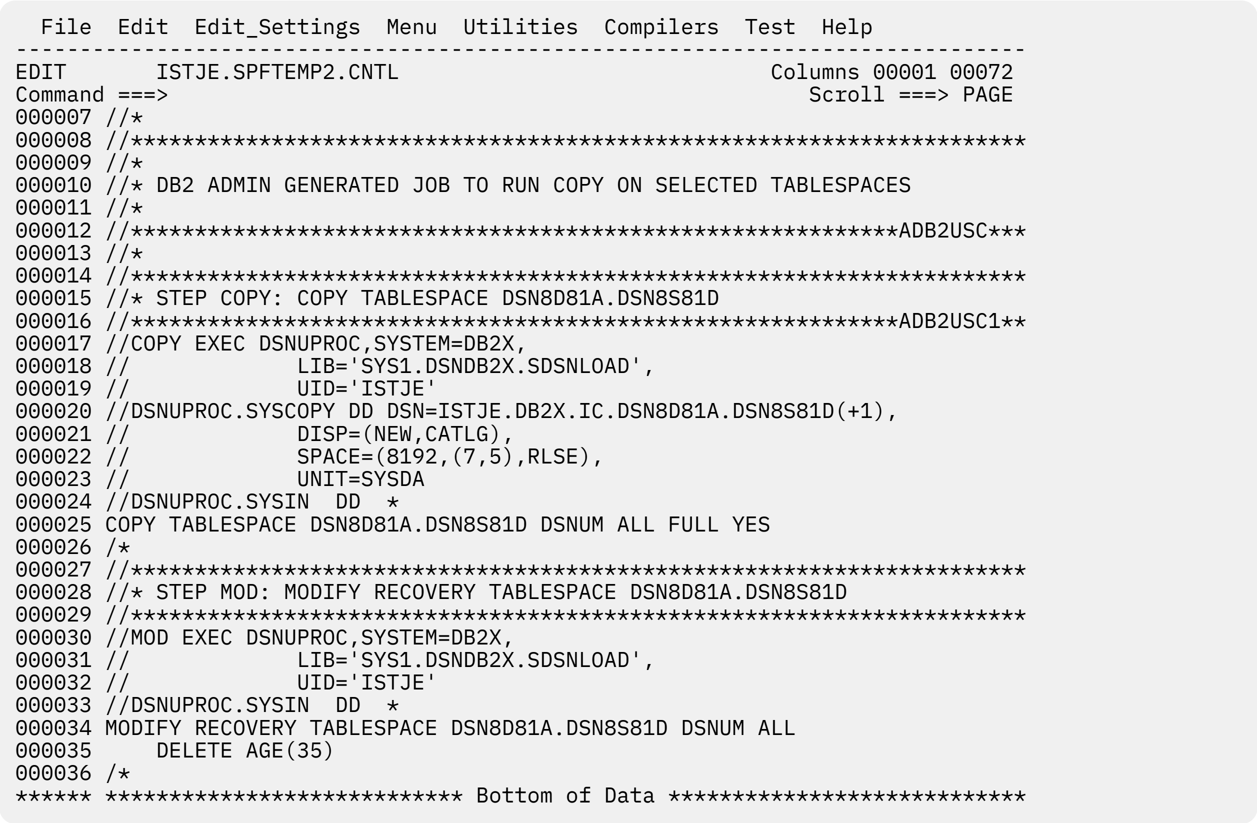
   Note: The **LC** option is displayed only when all of the following conditions are true:

   - The table does not contain XML columns.
   - The panel is displayed for one table space.
   - The table space contains only one table.
   - The table space is not a LOB table space.
   - The target table does not contain GENERATED ALWAYS columns.

   Figure 113. ***Table Space Utilities (ADB2US)*** *panel*

3. Specify option C to indicate that you want to use the COPY utility to take a full image copy of the table space.

   The generated JCL to run the utility job is displayed:



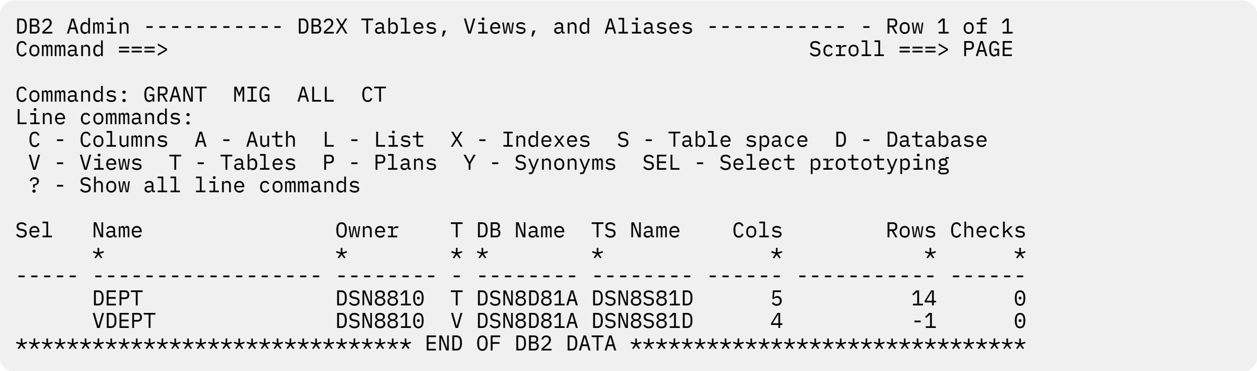
```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
-------------------------------------------------------------------------------
EDIT       ISTJE.SPFTEMP2.CNTL                             Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
000007 //*
000008 //**********************************************************************
000009 //*
000010 //* DB2 ADMIN GENERATED JOB TO RUN COPY ON SELECTED TABLESPACES
000011 //*
000012 //************************************************************ADB2USC***
000013 //*
000014 //**********************************************************************
000015 //* STEP COPY: COPY TABLESPACE DSN8D81A.DSN8S81D
000016 //************************************************************ADB2USC1**
000017 //COPY EXEC DSNUPROC,SYSTEM=DB2X,
000018 //             LIB='SYS1.DSNDB2X.SDSNLOAD',
000019 //             UID='ISTJE'
000020 //DSNUPROC.SYSCOPY DD DSN=ISTJE.DB2X.IC.DSN8D81A.DSN8S81D(+1),
000021 //             DISP=(NEW,CATLG),
000022 //             SPACE=(8192,(7,5),RLSE),
000023 //             UNIT=SYSDA
000024 //DSNUPROC.SYSIN  DD  *
000025 COPY TABLESPACE DSN8D81A.DSN8S81D DSNUM ALL FULL YES
000026 /*
000027 //**********************************************************************
000028 //* STEP MOD: MODIFY RECOVERY TABLESPACE DSN8D81A.DSN8S81D
000029 //**********************************************************************
000030 //MOD EXEC DSNUPROC,SYSTEM=DB2X,
000031 //             LIB='SYS1.DSNDB2X.SDSNLOAD',
000032 //             UID='ISTJE'
000033 //DSNUPROC.SYSIN  DD  *
000034 MODIFY RECOVERY TABLESPACE DSN8D81A.DSN8S81D DSNUM ALL
000035     DELETE AGE(35)
000036 /*
****** **************************** Bottom of Data ****************************
```

Figure 114. JCL for a utility

You can submit the job and then check the output to ensure that the utility completed successfully.

4. Exit back to the **Table Spaces (ADB21S)** panel.
5. Specify the T line command against table space DSN8S81D to see the tables that belong to that table space, and press Enter.

   The **Tables, Views, and Aliases (ADB21T)** panel shows the tables in DSN8S81D:

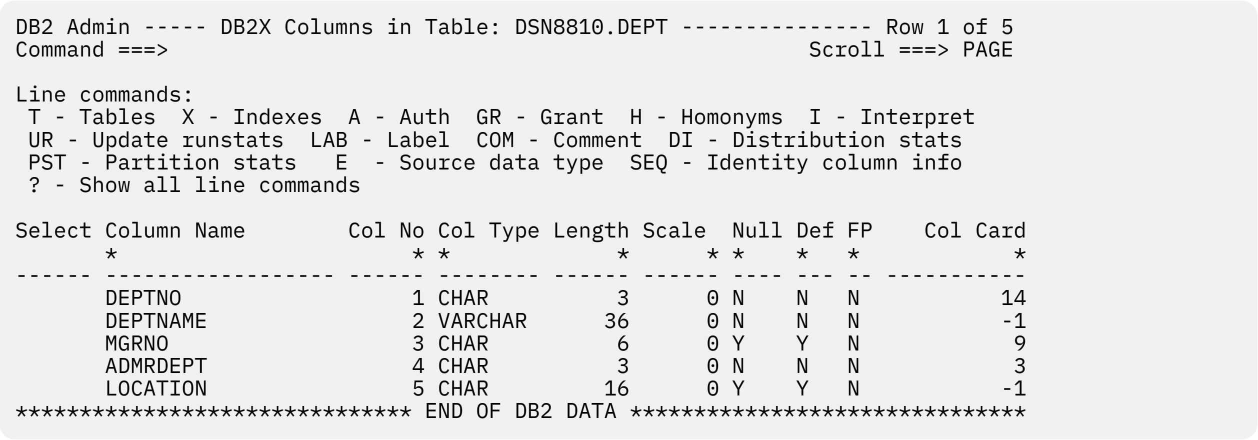


```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- - Row 1 of 1
Command ===>                                                  Scroll ===> PAGE

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

Sel   Name               Owner    T DB Name  TS Name    Cols        Rows Checks
      *                  *        * *        *             *           *      *
----- ------------------ -------- - -------- -------- ------ ----------- ------
      DEPT               DSN8810  T DSN8D81A DSN8S81D      5          14      0
      VDEPT              DSN8810  V DSN8D81A DSN8S81D      4          -1      0
******************************* END OF DB2 DATA *******************************
```

Figure 115. Tables in a table space (ADB21T)

6. Specify the C line command against the DEPT table to view the columns, and press Enter.

   The **Columns in Table (ADB21TC)** panel is displayed:

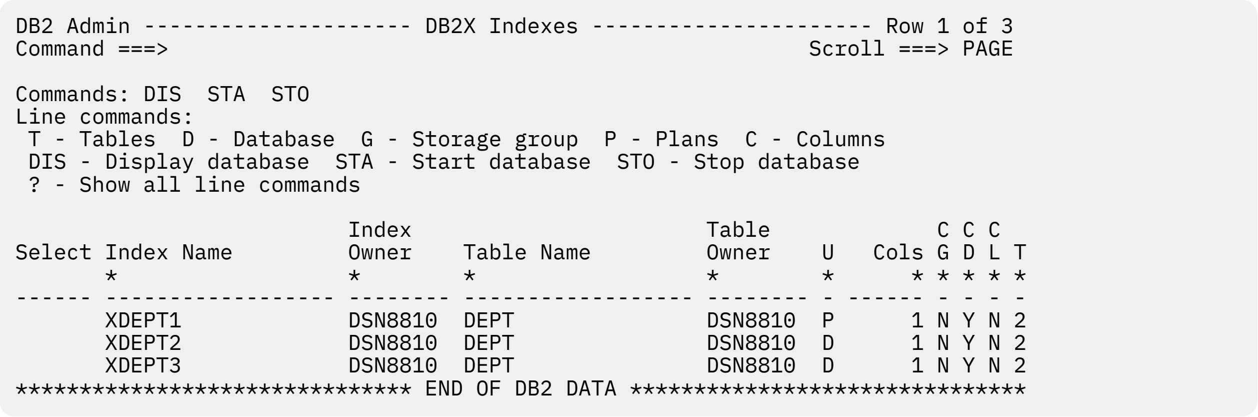


```
DB2 Admin ----- DB2X Columns in Table: DSN8810.DEPT --------------- Row 1 of 5
Command ===>                                                 Scroll ===> PAGE

Line commands:
 T - Tables  X - Indexes  A - Auth  GR - Grant  H - Homonyms  I - Interpret
 UR - Update runstats  LAB - Label  COM - Comment  DI - Distribution stats
 PST - Partition stats   E  - Source data type  SEQ - Identity column info
 ? - Show all line commands

Select Column Name       Col No Col Type Length Scale  Null Def FP    Col Card
       *                      * *             *      * *    *   *           *
------ ----------------- ------ -------- ------ ------ ---- --- -- -----------
       DEPTNO                 1 CHAR          3      0 N    N   N           14
       DEPTNAME               2 VARCHAR      36      0 N    N   N           -1
       MGRNO                  3 CHAR          6      0 Y    Y   N            9
       ADMRDEPT               4 CHAR          3      0 N    N   N            3
       LOCATION               5 CHAR         16      0 Y    Y   N           -1
******************************* END OF DB2 DATA ******************************
```

Figure 116. Columns in a table (ADB21TC)

7. Exit back to the **Tables, Views, and Aliases (ADB21T)** panel.
8. Specify the X line command against the DEPT table to view the indexes, and press Enter.

   The **Indexes (ADB21X)** panel is displayed:



```
DB2 Admin --------------------- DB2X Indexes --------------------- Row 1 of 3
Command ===>                                                 Scroll ===> PAGE

Commands: DIS  STA  STO
Line commands:
 T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
 DIS - Display database  STA - Start database  STO - Stop database
 ? - Show all line commands

                          Index                        Table             C C C
Select Index Name         Owner    Table Name          Owner    U   Cols G D L T
       *                  *        *                  *        *      * * * * *
------ ------------------ -------- ------------------ -------- - ------ - - - -
       XDEPT1             DSN8810  DEPT               DSN8810  P      1 N Y N 2
       XDEPT2             DSN8810  DEPT               DSN8810  D      1 N Y N 2
       XDEPT3             DSN8810  DEPT               DSN8810  D      1 N Y N 2
******************************* END OF DB2 DATA ******************************
```

Figure 117. Indexes for a table (ADB21X)

What to do next

Exit back to the **System Catalog (ADB21)** panel to complete the next stage of the tutorial.

Granting authorizations

You can use Db2 Admin Tool to grant authorizations on Db2 objects.

Before you begin

Ensure that the **System Catalog (ADB21)** panel is displayed.

About this task

This tutorial shows you how to do the following tasks:

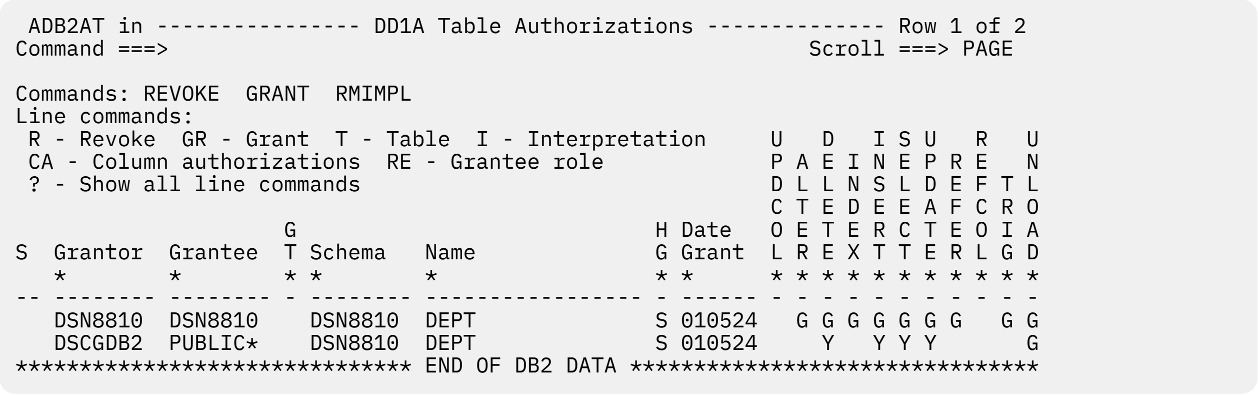
- View authorizations for a table.
- Grant authorizations for a table.

Restriction: The GRANT USAGE ON JAR statement is not supported in the IBM Db2 Object Comparison Tool for z/OS.

Procedure

1. On the **System Catalog (ADB21)** panel, specify option S, and press Enter.
2. On the **Table Spaces (ADB21S)** panel, specify the T line command against the table space that contains the DEPT table, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, specify the A line command against the DEPT table, and press Enter.

   The **Table Authorizations (ADB2AT)** panel shows the authorizations for the DEPT table:

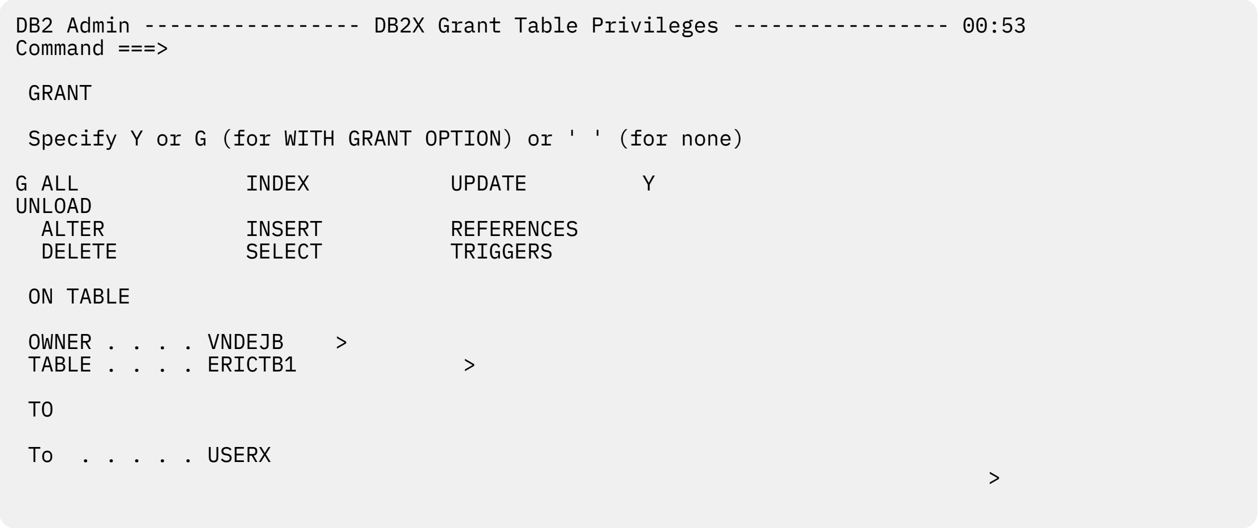


```
 ADB2AT in ---------------- DD1A Table Authorizations -------------- Row 1 of 2
Command ===>                                                  Scroll ===> PAGE

Commands: REVOKE  GRANT  RMIMPL
Line commands:
 R - Revoke  GR - Grant  T - Table  I - Interpretation     U   D   I S U   R   U
 CA - Column authorizations  RE - Grantee role             P A E I N E P R E   N
 ? - Show all line commands                                D L L N S L D E F T L
                                                           C T E D E E A F C R O
                     G                            H Date   O E T E R C T E O I A
S  Grantor  Grantee  T Schema   Name              G Grant  L R E X T T E R L G D
   *        *        * *        *                 * *      * * * * * * * * * * *
-- -------- -------- - -------- ----------------- - ------ - - - - - - - - - - -
   DSN8810  DSN8810    DSN8810  DEPT              S 010524   G G G G G G G   G G
   DSCGDB2  PUBLIC*    DSN8810  DEPT              S 010524     Y   Y Y Y       G
******************************* END OF DB2 DATA ********************************
```

   Note: The UNLOAD privilege is displayed only if you are running Db2 12 for z/OS.

   Figure 118. ***Table Authorizations (ADB2AT)*** *panel*

4. For the row with DSN8810 in the Grantor column, specify the GR line command in the **S** column, and press Enter.
5. On the **Grant Table Privileges (ADB2GT)** panel, specify Y or G next to the privileges that you want to grant, and press Enter.



```
DB2 Admin ----------------- DB2X Grant Table Privileges ----------------- 00:53
Command ===>

 GRANT

 Specify Y or G (for WITH GRANT OPTION) or ' ' (for none)

G ALL             INDEX           UPDATE         Y
UNLOAD
  ALTER           INSERT          REFERENCES
  DELETE          SELECT          TRIGGERS

 ON TABLE

 OWNER . . . . VNDEJB    >
 TABLE . . . . ERICTB1             >

 TO

 To  . . . . . USERX
                                                                            >
```

   Note: The UNLOAD privilege is displayed only if you are running Db2 12 for z/OS.

   Figure 119. ***Grant Table Privileges (ADB2GT)*** *panel*

What to do next

Exit back to the **System Catalog (ADB21)** panel to complete the next stage of the tutorial.

Binding plans and packages

You can bind plans and packages in Db2 Admin Tool.

Before you begin

Ensure that the **System Catalog (ADB21)** panel is displayed.

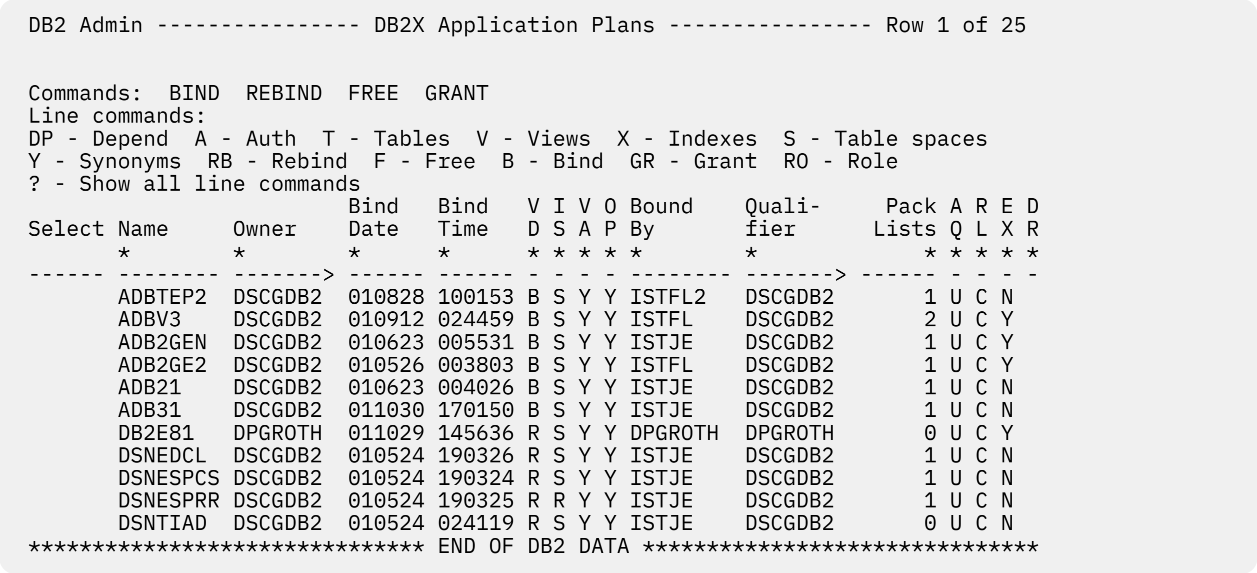
About this task

In this stage of the tutorial, you will view a list of application plans in the system catalog and bind a plan.

Procedure

1. Specify option P, and press Enter to list the application plans in the catalog.

   The **Application Plans (ADB21P)** panel is displayed:



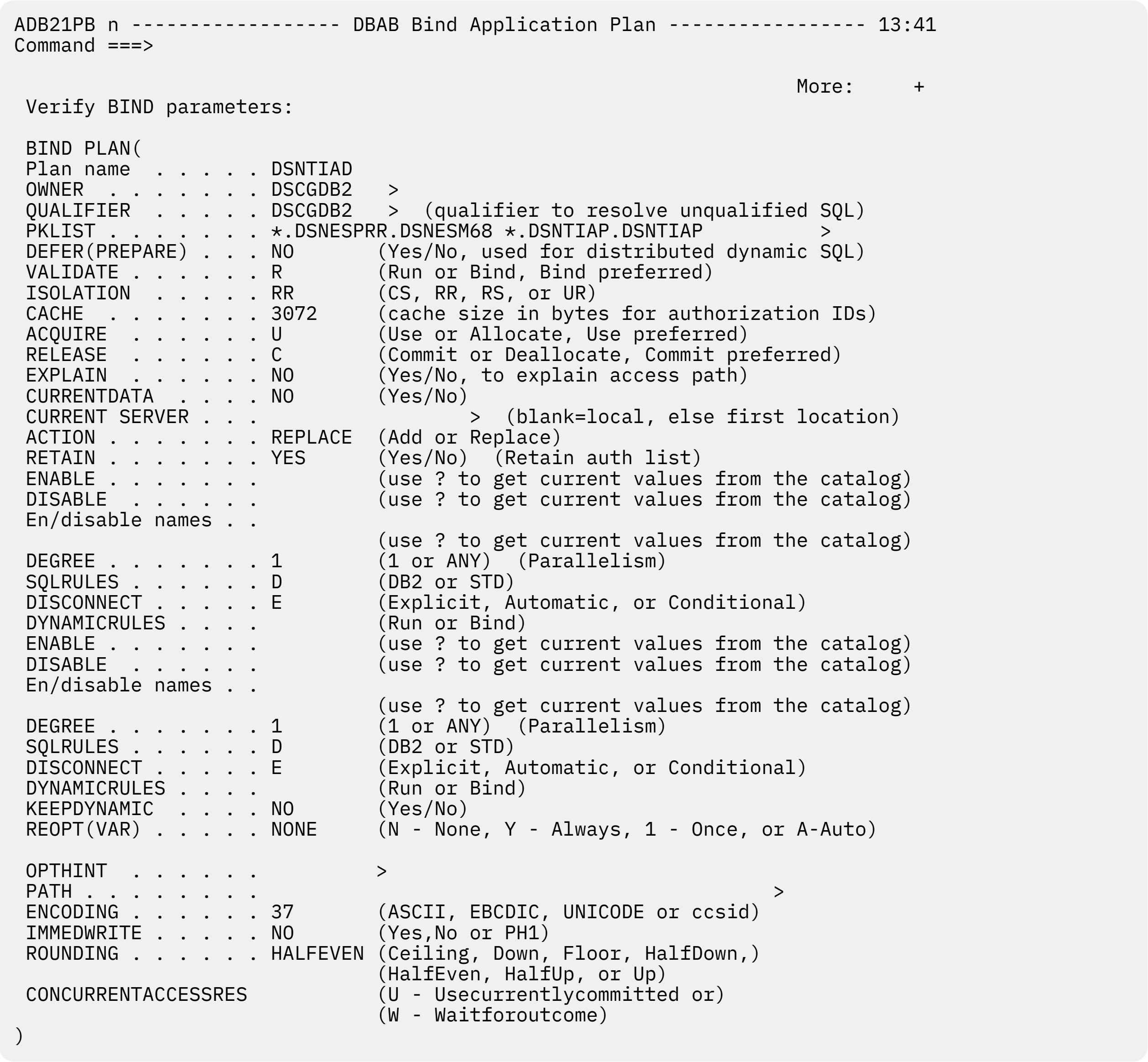
```
DB2 Admin ---------------- DB2X Application Plans ---------------- Row 1 of 25


Commands:  BIND  REBIND  FREE  GRANT
Line commands:
DP - Depend  A - Auth  T - Tables  V - Views  X - Indexes  S - Table spaces
Y - Synonyms  RB - Rebind  F - Free  B - Bind  GR - Grant  RO - Role
? - Show all line commands
                         Bind   Bind   V I V O Bound    Quali-     Pack A R E D
Select Name     Owner    Date   Time   D S A P By       fier      Lists Q L X R
       *        *        *      *      * * * * *        *              * * * * *
------ -------- -------> ------ ------ - - - - -------- -------> ------ - - - -
       ADBTEP2  DSCGDB2  010828 100153 B S Y Y ISTFL2   DSCGDB2       1 U C N
       ADBV3    DSCGDB2  010912 024459 B S Y Y ISTFL    DSCGDB2       2 U C Y
       ADB2GEN  DSCGDB2  010623 005531 B S Y Y ISTJE    DSCGDB2       1 U C Y
       ADB2GE2  DSCGDB2  010526 003803 B S Y Y ISTFL    DSCGDB2       1 U C Y
       ADB21    DSCGDB2  010623 004026 B S Y Y ISTJE    DSCGDB2       1 U C N
       ADB31    DSCGDB2  011030 170150 B S Y Y ISTJE    DSCGDB2       1 U C N
       DB2E81   DPGROTH  011029 145636 R S Y Y DPGROTH  DPGROTH       0 U C Y
       DSNEDCL  DSCGDB2  010524 190326 R S Y Y ISTJE    DSCGDB2       1 U C N
       DSNESPCS DSCGDB2  010524 190324 R S Y Y ISTJE    DSCGDB2       1 U C N
       DSNESPRR DSCGDB2  010524 190325 R R Y Y ISTJE    DSCGDB2       1 U C N
       DSNTIAD  DSCGDB2  010524 024119 R S Y Y ISTJE    DSCGDB2       0 U C N
******************************* END OF DB2 DATA *******************************
```

*Figure 120. **Application Plans (ADB21P)** panel*

   On this panel, you can issue a B (Bind), RB (Rebind), or F (Free) line command for a particular plan. You can also issue a BIND, REBIND, FREE, or GRANT primary command for all plans that are listed.

2. For example, specify B against plan DSNTIAD, and press Enter to bind this plan.

   The **Bind Application Plan (ADB21PB)** panel is displayed:



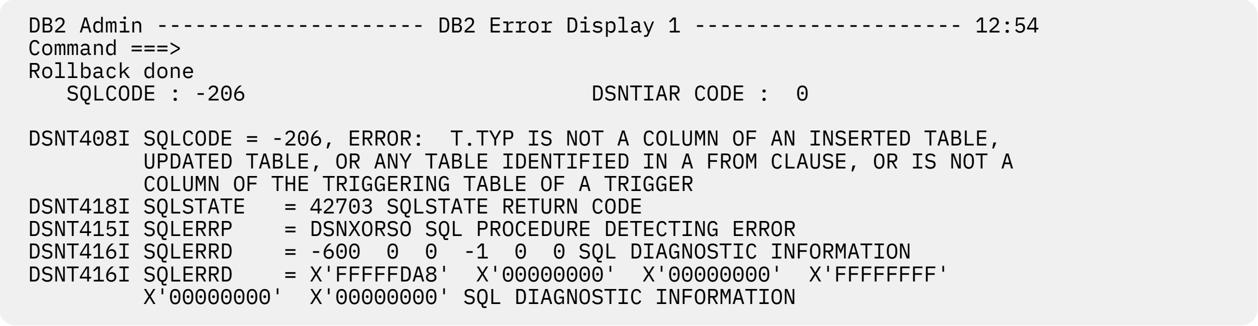
```
ADB21PB n ------------------ DBAB Bind Application Plan ----------------- 13:41
Command ===>
                                                                   More:     +
 Verify BIND parameters:

 BIND PLAN(
 Plan name  . . . . . DSNTIAD
 OWNER  . . . . . . . DSCGDB2   >
 QUALIFIER  . . . . . DSCGDB2   >  (qualifier to resolve unqualified SQL)
 PKLIST . . . . . . . *.DSNESPRR.DSNESM68 *.DSNTIAP.DSNTIAP          >
 DEFER(PREPARE) . . . NO       (Yes/No, used for distributed dynamic SQL)
 VALIDATE . . . . . . R        (Run or Bind, Bind preferred)
 ISOLATION  . . . . . RR       (CS, RR, RS, or UR)
 CACHE  . . . . . . . 3072     (cache size in bytes for authorization IDs)
 ACQUIRE  . . . . . . U        (Use or Allocate, Use preferred)
 RELEASE  . . . . . . C        (Commit or Deallocate, Commit preferred)
 EXPLAIN  . . . . . . NO       (Yes/No, to explain access path)
 CURRENTDATA  . . . . NO       (Yes/No)
 CURRENT SERVER . . .                  >  (blank=local, else first location)
 ACTION . . . . . . . REPLACE  (Add or Replace)
 RETAIN . . . . . . . YES      (Yes/No)  (Retain auth list)
 ENABLE . . . . . . .          (use ? to get current values from the catalog)
 DISABLE  . . . . . .          (use ? to get current values from the catalog)
 En/disable names . .
                               (use ? to get current values from the catalog)
 DEGREE . . . . . . . 1        (1 or ANY)  (Parallelism)
 SQLRULES . . . . . . D        (DB2 or STD)
 DISCONNECT . . . . . E        (Explicit, Automatic, or Conditional)
 DYNAMICRULES . . . .          (Run or Bind)
 ENABLE . . . . . . .          (use ? to get current values from the catalog)
 DISABLE  . . . . . .          (use ? to get current values from the catalog)
 En/disable names . .
                               (use ? to get current values from the catalog)
 DEGREE . . . . . . . 1        (1 or ANY)  (Parallelism)
 SQLRULES . . . . . . D        (DB2 or STD)
 DISCONNECT . . . . . E        (Explicit, Automatic, or Conditional)
 DYNAMICRULES . . . .          (Run or Bind)
 KEEPDYNAMIC  . . . . NO       (Yes/No)
 REOPT(VAR) . . . . . NONE     (N - None, Y - Always, 1 - Once, or A-Auto)

 OPTHINT  . . . . . .          >
 PATH . . . . . . . .                                              >
 ENCODING . . . . . . 37       (ASCII, EBCDIC, UNICODE or ccsid)
 IMMEDWRITE . . . . . NO       (Yes,No or PH1)
 ROUNDING . . . . . . HALFEVEN (Ceiling, Down, Floor, HalfDown,)
                               (HalfEven, HalfUp, or Up)
 CONCURRENTACCESSRES           (U - Usecurrentlycommitted or)
                               (W - Waitforoutcome)
)
```

Figure 121. ***Bind Application Plan (ADB21PB)*** *panel*

3. Press Enter to verify the BIND parameters.

   Db2 Admin Tool uses the catalog to find the DBRM members and libraries for the bind operation.

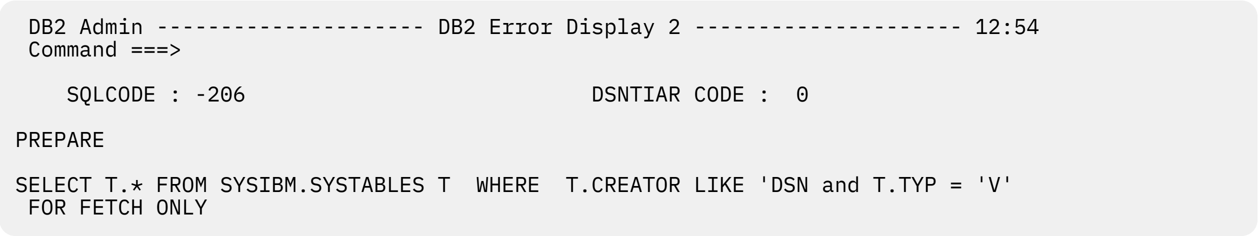
   If an SQL error occurs, Db2 Admin Tool displays the DSNTIAR message:



```
 DB2 Admin --------------------- DB2 Error Display 1 --------------------- 12:54
 Command ===>
 Rollback done
    SQLCODE : -206                          DSNTIAR CODE :  0

 DSNT408I SQLCODE = -206, ERROR:  T.TYP IS NOT A COLUMN OF AN INSERTED TABLE,
          UPDATED TABLE, OR ANY TABLE IDENTIFIED IN A FROM CLAUSE, OR IS NOT A
          COLUMN OF THE TRIGGERING TABLE OF A TRIGGER
 DSNT418I SQLSTATE   = 42703 SQLSTATE RETURN CODE
 DSNT415I SQLERRP    = DSNXORSO SQL PROCEDURE DETECTING ERROR
 DSNT416I SQLERRD    = -600  0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
 DSNT416I SQLERRD    = X'FFFFFDA8'  X'00000000'  X'00000000'  X'FFFFFFFF'
          X'00000000'  X'00000000' SQL DIAGNOSTIC INFORMATION
```

Figure 122. DSNTIAR error messages

When you press Enter, a second error panel opens to display the SQL statement that caused the error:



```
 DB2 Admin --------------------- DB2 Error Display 2 --------------------- 12:54
 Command ===>

    SQLCODE : -206                           DSNTIAR CODE :  0

PREPARE

SELECT T.* FROM SYSIBM.SYSTABLES T  WHERE  T.CREATOR LIKE 'DSN and T.TYP = 'V'
 FOR FETCH ONLY
```

Figure 123. SQL statement in error

What to do next

Return to the **Application Plans (ADB21P)** panel.

Displaying detailed information about an object

You can display detailed information about an object in Db2 Admin Tool.

Before you begin

Ensure that you are on the **Application Plans** panel.

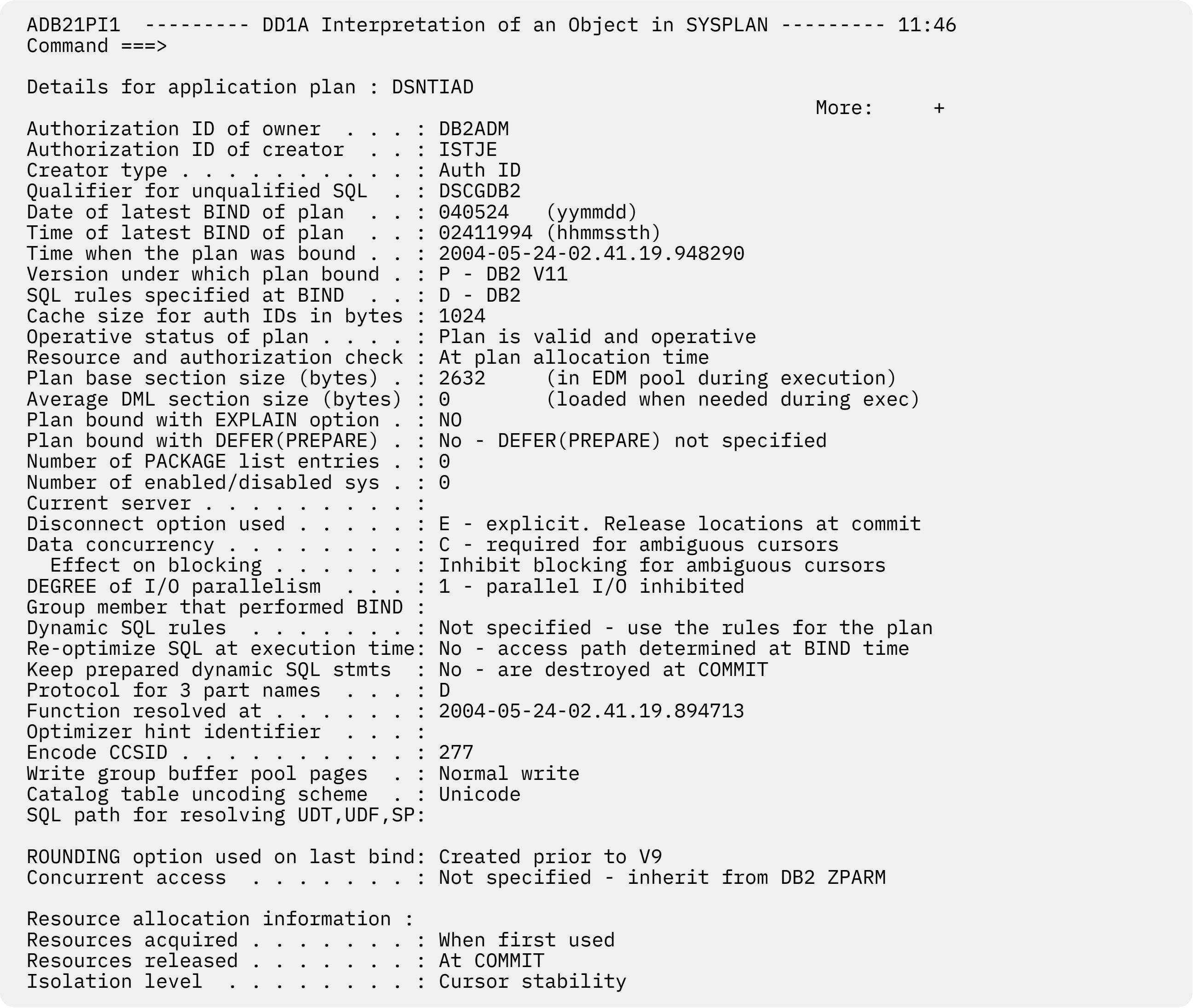
About this task

In this stage of the tutorial, you will view details about an application plan.

Procedure

Specify the I line command against application plan DSNTIAD, and press Enter.

The **Interpretation of an Object** panel is displayed, as shown in the following figure.



```
ADB21PI1  --------- DD1A Interpretation of an Object in SYSPLAN --------- 11:46
Command ===>

Details for application plan : DSNTIAD
                                                                  More:     +
Authorization ID of owner  . . . : DB2ADM
Authorization ID of creator  . . : ISTJE
Creator type . . . . . . . . . . : Auth ID
Qualifier for unqualified SQL  . : DSCGDB2
Date of latest BIND of plan  . . : 040524   (yymmdd)
Time of latest BIND of plan  . . : 02411994 (hhmmssth)
Time when the plan was bound . . : 2004-05-24-02.41.19.948290
Version under which plan bound . : P - DB2 V11
SQL rules specified at BIND  . . : D - DB2
Cache size for auth IDs in bytes : 1024
Operative status of plan . . . . : Plan is valid and operative
Resource and authorization check : At plan allocation time
Plan base section size (bytes) . : 2632     (in EDM pool during execution)
Average DML section size (bytes) : 0        (loaded when needed during exec)
Plan bound with EXPLAIN option . : NO
Plan bound with DEFER(PREPARE) . : No - DEFER(PREPARE) not specified
Number of PACKAGE list entries . : 0
Number of enabled/disabled sys . : 0
Current server . . . . . . . . . :
Disconnect option used . . . . . : E - explicit. Release locations at commit
Data concurrency . . . . . . . . : C - required for ambiguous cursors
  Effect on blocking . . . . . . : Inhibit blocking for ambiguous cursors
DEGREE of I/O parallelism  . . . : 1 - parallel I/O inhibited
Group member that performed BIND :
Dynamic SQL rules  . . . . . . . : Not specified - use the rules for the plan
Re-optimize SQL at execution time: No - access path determined at BIND time
Keep prepared dynamic SQL stmts  : No - are destroyed at COMMIT
Protocol for 3 part names  . . . : D
Function resolved at . . . . . . : 2004-05-24-02.41.19.894713
Optimizer hint identifier  . . . :
Encode CCSID . . . . . . . . . . : 277
Write group buffer pool pages  . : Normal write
Catalog table uncoding scheme  . : Unicode
SQL path for resolving UDT,UDF,SP:

ROUNDING option used on last bind: Created prior to V9
Concurrent access  . . . . . . . : Not specified - inherit from DB2 ZPARM

Resource allocation information :
Resources acquired . . . . . . . : When first used
Resources released . . . . . . . : At COMMIT
Isolation level  . . . . . . . . : Cursor stability
```

Figure 124. Interpretation of an object (ADB21PI1)

What to do next

Return to the **System Catalog** panel to complete the next stage of the tutorial.

Reverse engineering objects

You can extract from the Db2 catalog the DDL that is required to re-create Db2 objects. This process is called *reverse engineering* the objects.

Before you begin

Ensure that the **System Catalog (ADB21)** panel is displayed.

About this task

The starting point for reverse engineering can be databases, table spaces, tables, aliases, synonyms, schemas, data types, functions, stored procedures, triggers, sequences, or storage groups.

In this stage of the tutorial, you will complete the following steps:

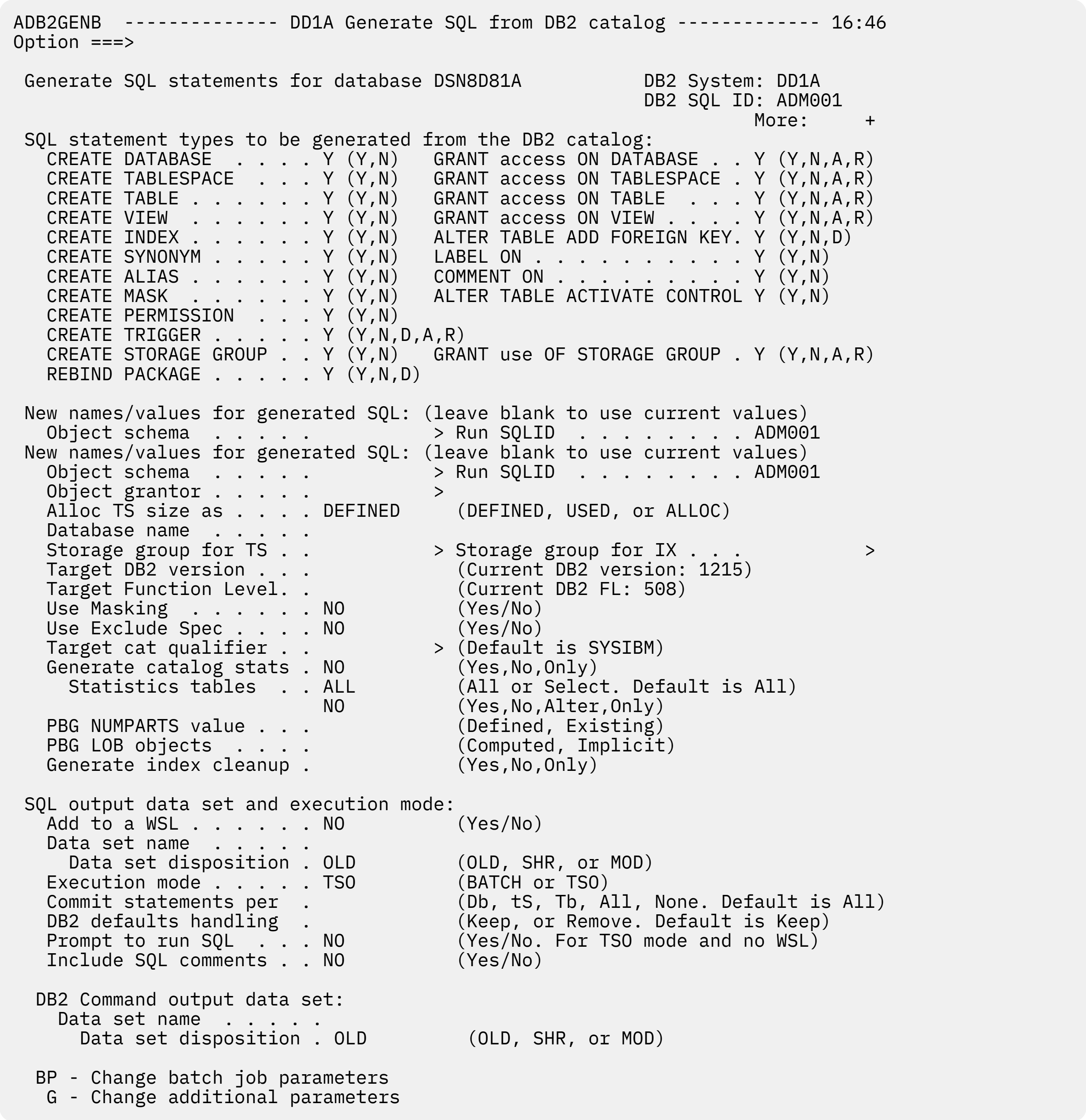
- Reverse engineer a database.
- View the output of reverse engineering that database.

- Issue a rebind and view the output.

Procedure

1. On the **System Catalog (ADB21)** panel, specify option D, and press Enter.
2. On the **Databases (ADB21D)** panel, specify the GEN line command against the DSN8D81A database, and press Enter.

   The **Generate SQL from DB2 catalog (ADB2GENB)** panel is displayed, as shown in the following figure:



```
ADB2GENB  -------------- DD1A Generate SQL from DB2 catalog ------------- 16:46
Option ===>

 Generate SQL statements for database DSN8D81A           DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
                                                                   More:     +
 SQL statement types to be generated from the DB2 catalog:
   CREATE DATABASE  . . . . Y (Y,N)   GRANT access ON DATABASE . . Y (Y,N,A,R)
   CREATE TABLESPACE  . . . Y (Y,N)   GRANT access ON TABLESPACE . Y (Y,N,A,R)
   CREATE TABLE . . . . . . Y (Y,N)   GRANT access ON TABLE  . . . Y (Y,N,A,R)
   CREATE VIEW  . . . . . . Y (Y,N)   GRANT access ON VIEW . . . . Y (Y,N,A,R)
   CREATE INDEX . . . . . . Y (Y,N)   ALTER TABLE ADD FOREIGN KEY. Y (Y,N,D)
   CREATE SYNONYM . . . . . Y (Y,N)   LABEL ON . . . . . . . . . . Y (Y,N)
   CREATE ALIAS . . . . . . Y (Y,N)   COMMENT ON . . . . . . . . . Y (Y,N)
   CREATE MASK  . . . . . . Y (Y,N)   ALTER TABLE ACTIVATE CONTROL Y (Y,N)
   CREATE PERMISSION  . . . Y (Y,N)
   CREATE TRIGGER . . . . . Y (Y,N,D,A,R)
   CREATE STORAGE GROUP . . Y (Y,N)   GRANT use OF STORAGE GROUP . Y (Y,N,A,R)
   REBIND PACKAGE . . . . . Y (Y,N,D)

 New names/values for generated SQL: (leave blank to use current values)
   Object schema  . . . . .           > Run SQLID  . . . . . . . . ADM001
 New names/values for generated SQL: (leave blank to use current values)
   Object schema  . . . . .           > Run SQLID  . . . . . . . . ADM001
   Object grantor . . . . .           >
   Alloc TS size as . . . . DEFINED     (DEFINED, USED, or ALLOC)
   Database name  . . . . .
   Storage group for TS . .           > Storage group for IX . . .           >
   Target DB2 version . . .             (Current DB2 version: 1215)
   Target Function Level. .             (Current DB2 FL: 508)
   Use Masking  . . . . . . NO          (Yes/No)
   Use Exclude Spec . . . . NO          (Yes/No)
   Target cat qualifier . .           > (Default is SYSIBM)
   Generate catalog stats . NO          (Yes,No,Only)
     Statistics tables  . . ALL         (All or Select. Default is All)
                            NO          (Yes,No,Alter,Only)
   PBG NUMPARTS value . . .             (Defined, Existing)
   PBG LOB objects  . . . .             (Computed, Implicit)
   Generate index cleanup .             (Yes,No,Only)

 SQL output data set and execution mode:
   Add to a WSL . . . . . . NO          (Yes/No)
   Data set name  . . . . .
     Data set disposition . OLD         (OLD, SHR, or MOD)
   Execution mode . . . . . TSO         (BATCH or TSO)
   Commit statements per  .             (Db, tS, Tb, All, None. Default is All)
   DB2 defaults handling  .             (Keep, or Remove. Default is Keep)
   Prompt to run SQL  . . . NO          (Yes/No. For TSO mode and no WSL)
   Include SQL comments . . NO          (Yes/No)

  DB2 Command output data set:
    Data set name  . . . . .
      Data set disposition . OLD         (OLD, SHR, or MOD)

  BP - Change batch job parameters
   G - Change additional parameters
```

   Figure 125. ***Generate SQL from DB2 catalog (ADB2GENB)*** *panel*

3. To view the output, press Enter.

   The following figure shows part of the result of reverse engineering this database:

```
------------------------------------------------------------------------
--                                                                    --
-- Database 2 Administration Tool (DB2 Admin) , program 5655-DAT (C)  --
--                                                                    --
-- ADB2GEN  -  Extract object definitions from the DB2 Catalog tables --
--                                                                    --
-- Input prepared on : DB2X (810)     Extract time : 2013-16-04 01:01 --
--                                                                    --
-- Catalog values overridden : none                                   --
--                                                                    --
-- Generate : SG=Y DB=Y TS=Y TB=Y VW=Y IX=Y SY=Y AL=Y LB=Y CM=Y FK=Y  --
--            TG=Y UT=N UF=N SP=N                                     --
-- Grants   : SG=Y DB=Y TS=Y TB=Y VW=Y SC=N UT=N UF=N SP=N            --
--                                                                    --
------------------------------------------------------------------------
------------------------------------------------------------------------
--                                                                    --
-- ADB2GEN: Generate DDL for Database DSN8D81A                        --
--                                                                    --
------------------------------------------------------------------------
--
--
------------------------------------------------------------------------
-- Database=DSN8D81A   Stogroup=DSN8G810
------------------------------------------------------------------------
--
  SET CURRENT SQLID='DSCGDB2';
--
  CREATE DATABASE DSN8D81A
    BUFFERPOOL BP0
    INDEXBP    BP2
    CCSID      EBCDIC
    STOGROUP DSN8G810 ;
--
  GRANT DBADM
    ON DATABASE DSN8D81A TO PUBLIC;
--
  COMMIT;
--
```

Figure 126. Reverse engineering output (1 of 2)

```
------------------------------------------------------------------------
-- Database=DSN8D81A   Stogroup=DSN8G810
-- Tablespace=DSN8D81A.DSN8S81D
------------------------------------------------------------------------
--
  CREATE TABLESPACE DSN8S81D
    IN DSN8D81A
    USING STOGROUP DSN8G810
    PRIQTY 32 SECQTY 20
    ERASE  NO
    FREEPAGE 0 PCTFREE 5
    GBPCACHE CHANGED
    TRACKMOD YES
    BUFFERPOOL BP0
    LOCKSIZE PAGE
    LOCKMAX SYSTEM
    CLOSE NO
    COMPRESS NO
    CCSID      EBCDIC
    MAXROWS 255;
--
  GRANT USE OF TABLESPACE DSN8D81A.DSN8S81D TO PUBLIC;
--
  COMMIT;
--
------------------------------------------------------------------------
--    Table=DSN8810.DEPT                In DSN8D81A.DSN8S81D
------------------------------------------------------------------------
--
  SET CURRENT SQLID='DSN8810';
--
  CREATE TABLE DSN8810.DEPT
     (DEPTNO               CHAR(3) FOR SBCS DATA NOT NULL ,
      DEPTNAME             VARCHAR(36) FOR SBCS DATA NOT NULL ,
      MGRNO                CHAR(6) FOR SBCS DATA WITH DEFAULT NULL ,
      ADMRDEPT             CHAR(3) FOR SBCS DATA NOT NULL ,
```

Figure 127. Reverse engineering output (2 of 2)

4. Return to the **Generate SQL from DB2 catalog (ADB2GENB)** panel to choose to rebind the plan/ package and view the output.
5. In the **REBIND PLAN/PACKAGE** field, specify Y, and press Enter.

   The output is the same as shown in the previous figures and includes the following output:

```
Command ===>                                                  Scroll ===> PAGE
****** ***************************** Top of Data ******************************
000001   REBIND PACKAGE(DSN8ES81.DSN8ES1)
****** **************************** Bottom of Data ****************************
```

Figure 128. Reverse engineering rebind output

Chapter 4. Db2 management

The following information explains how to use Db2 Admin Tool features to simplify the way you manage your Db2 environments.

Querying the Db2 system catalog

You can use Db2 Admin Tool to navigate through object information in the Db2 catalog and perform various actions on the objects.

About this task

Specifically, you can perform the following tasks:

- Display any object in the Db2 catalog
- Display related Db2 objects
- Interpret catalog information
- Show the authorizations for Db2 objects
- Display the static SQL statements from application plans and packages
- Display the DDL for existing views
- Generate JCL (job control language) for the Db2 utilities and then run them online
- Execute dynamic SQL statements
- Issue Db2 commands for databases and database objects
- Display database structures
- Reverse engineer Db2 objects
- Generate reports about the Db2 objects that are saved in a printable format

Tip: To navigate directly to the Db2 catalog information for any object from any panel, use the ? command. For example, the following command displays the information for a database named MYDB:

```
?d MYDB
```

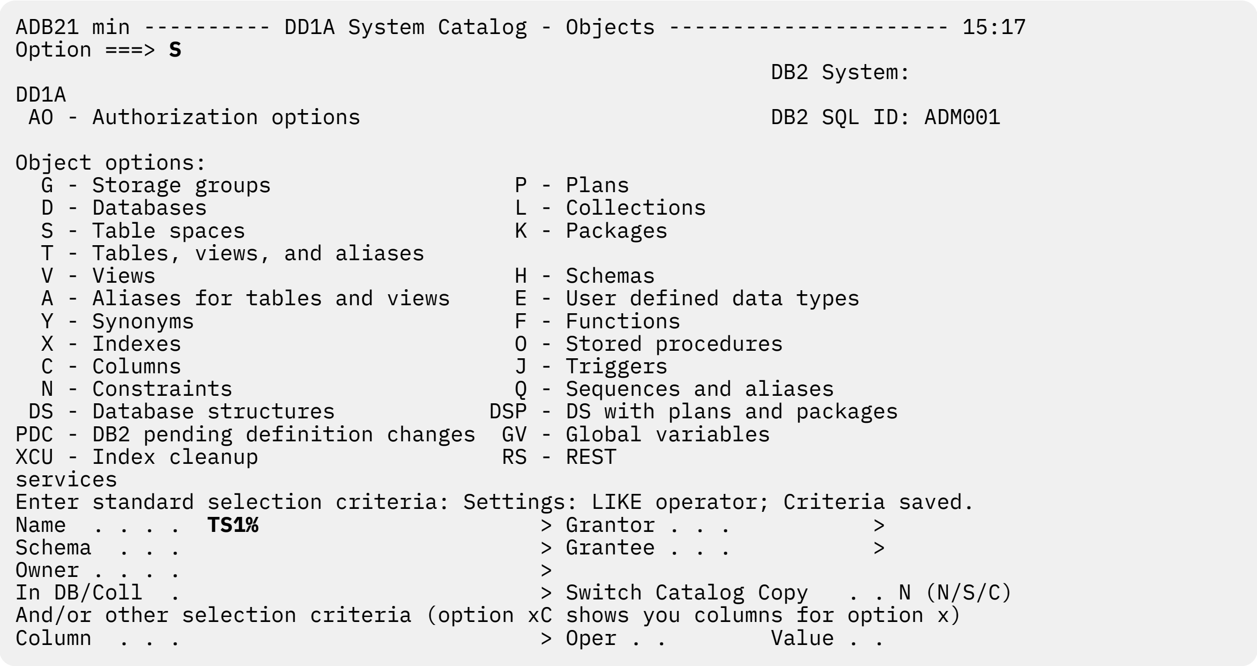
For more details about the ? command, see "Db2 Admin Tool primary commands" on page 211.

Procedure

To query the Db2 system catalog:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify an object and optionally any filtering criteria, and press Enter.

   For example, if you want to view all table spaces that start with TS1, specify the following information:



```
ADB21 min ---------- DD1A System Catalog - Objects --------------------- 15:17
Option ===> S
                                                           DB2 System:
DD1A
 AO - Authorization options                                DB2 SQL ID: ADM001

Object options:
  G - Storage groups                   P - Plans
  D - Databases                        L - Collections
  S - Table spaces                     K - Packages
  T - Tables, views, and aliases
  V - Views                            H - Schemas
  A - Aliases for tables and views     E - User defined data types
  Y - Synonyms                         F - Functions
  X - Indexes                          O - Stored procedures
  C - Columns                          J - Triggers
  N - Constraints                      Q - Sequences and aliases
 DS - Database structures            DSP - DS with plans and packages
PDC - DB2 pending definition changes  GV - Global variables
XCU - Index cleanup                   RS - REST
services
Enter standard selection criteria: Settings: LIKE operator; Criteria saved.
Name  . . . .  TS1%                      > Grantor . . .           >
Schema  . . .                            > Grantee . . .           >
Owner . . . .                            >
In DB/Coll  .                            > Switch Catalog Copy   . . N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .                            > Oper . .        Value . .
```

For more information about this panel, see "The System Catalog (ADB21) panel" on page 135.

Information about the requested object or objects is displayed. The specific panel that is displayed depends on the object. However each panel has commands, line commands, or both that you can use to perform various tasks. For more information about each object panel, see "System catalog panels" on page 135.

Switching copies of the Db2 catalog

If your subsystem supports using multiple copies of the Db2 catalog, you can switch between copies of the catalog when using Db2 Admin Tool.

Before you begin

Before you select a catalog copy, you might want to refresh the copy. To do so, request the owner of the copy run the CPYRUN*xx* job that was generated when the copy was made. See "Making copies of the Db2 catalog for Db2 Admin Tool" on page 1049.

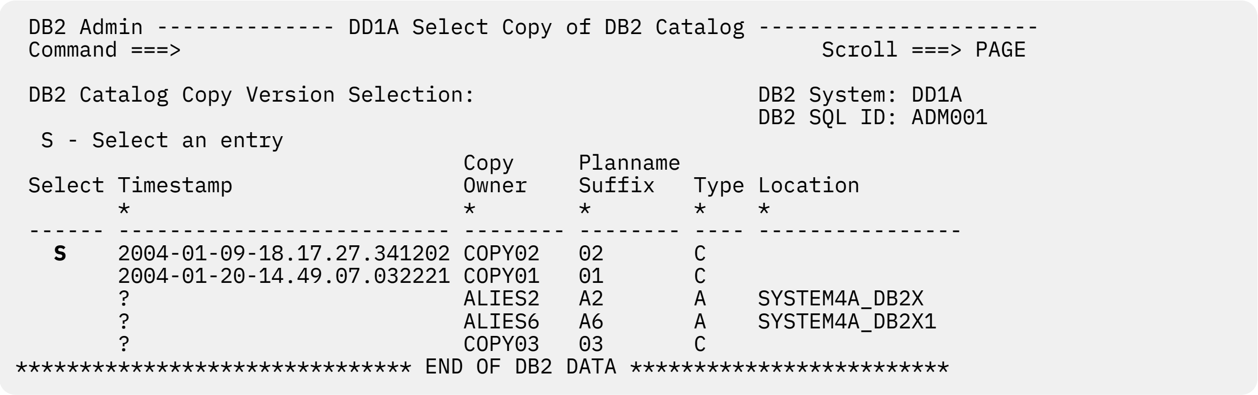
About this task

Recommendation: When using multiple copies of the catalog, do not issue requests that involve data for which the object definition has changed since the catalog copy was refreshed.

Procedure

To switch copies of the Db2 catalog:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, in the **Switch Catalog Copy** field, specify C to indicate that you want to use a copy of the catalog.
3. On the **Select Copy of DB2 Catalog (ADB2CCS)** panel, specify S in the Select column for the catalog copy that you want, and press Enter:



```
 DB2 Admin -------------- DD1A Select Copy of DB2 Catalog ----------------------
 Command ===>                                                  Scroll ===> PAGE

 DB2 Catalog Copy Version Selection:                      DB2 System: DD1A
                                                          DB2 SQL ID: ADM001
  S - Select an entry
                                   Copy     Planname
 Select Timestamp                  Owner    Suffix   Type Location
        *                          *        *        *    *
 ------ -------------------------- -------- -------- ---- ----------------
   S    2004-01-09-18.17.27.341202 COPY02   02       C
        2004-01-20-14.49.07.032221 COPY01   01       C
        ?                          ALIES2   A2       A    SYSTEM4A_DB2X
        ?                          ALIES6   A6       A    SYSTEM4A_DB2X1
        ?                          COPY03   03       C
******************************* END OF DB2 DATA *************************
```

When you use a copy of the Db2 catalog, the value in the **Planname Suffix** column is used on the header of the system catalog panels instead of the Db2 subsystem identifier.

Related concepts

“Copies of the Db2 catalog” on page 1049
You can create copies of the Db2 catalog to be used by Db2 Admin Tool.

Related reference

“The System Catalog (ADB21) panel” on page 135
Use the **System Catalog (ADB21)** panel to search for Db2 objects or authorizations in the Db2 catalog. To display this panel, select option 1 on the **DB2 Administration Menu (ADB2)** panel.

Reports from the Db2 catalog

You can use the REP command to generate reports that can be saved and printed.

When you use the system catalog panels to display information about the objects in the Db2 catalog, you can use the REP command to generate reports with information (that is similar to the displayed information) that can be saved and printed.

Reviewing printed reports can be faster than stepping through the information online. Saving reports about your databases at various points in time also allows you to perform trend analysis, which enables you to manage your environment more efficiently and more proactively.

When you issue the REP command, a panel is displayed that allows you to specify the content of the report. You choose which types of objects that you want included in the report. For example, for a database, you might want a report that lists the table spaces, tables, and indexes in the database. Or, for a group of schemas, you might want a report that lists the distinct types in each schema.

After you specify the objects for the report, Db2 Admin Tool generates JCL for a batch job that produces the report in a printable format. The batch job contains two steps. The first step invokes the GEN function to produce a version file for the objects that are to be included in the report. The second step formats the records in the version file into a report that is written to a data set.

The generated report consists of the following sections:

- A summary section that lists which types of objects are included in the report (the GEN parameters that were active when the data was collected).
- A detailed report section for each type of object that is included in the report. Each detailed report section lists all of the occurrences of the particular object. The information that is provided for each object and the column headings are the same as what is displayed on the corresponding system catalog panel for the object.

The following figure shows an example of the summary section of the report:

```
ADB2GEN parameters active when this data was collected :

Create Database(s)      : Yes   Create Tablespace(s)    : Yes   Create Table(s)         : Yes
Create View(s)          : No    Create Index(es)        : No    Create Synonym(s)       : No
Create Alias(es)        : No    Create Label(s)         : No
Create Triggers         : No    also for refs not gen'd : No
Create Foreign key(s)   : No    also for refs not gen'd : No
Create User def. Types  : No    Create Functions        : No    Create Stored Procedures: No

Column information will not be included in this report.
```

Figure 129. Example of the summary section

The following figure shows an example of a detailed report section for table spaces:

```
Name     DB Name   Parts Bpool  L E S I C Tables  Act. pages  Segsz T L
*        *            * *       * * * * *      *           *      * * *
-------- -------- ------ ------ - - - - - ------ ----------- ------ - -
SYSALTER DSNDB06       0 BP32K  P N A N N      2          44      4   Y
SYSCOPY  DSNDB06       0 BP0    A N A N N      2         720      0   Y
SYSDBASE DSNDB06       0 BP8K0  A N A N N     14        8280      0   Y
SYSDBAUT DSNDB06       0 BP0    A N A N N      4          84      0   Y
SYSDDF   DSNDB06       0 BP0    P N A N N      8          38      4   Y
SYSEBCDC DSNDB06       0 BP0    P N A N N      1          12      4   Y
SYSGPAUT DSNDB06       0 BP0    A N A N N      1         720      0   Y
SYSGROUP DSNDB06       0 BP0    A N A N N      2          24      0   Y
SYSGRTNS DSNDB06       0 BP8K0  R N A N N      2          24      4   Y
SYSHIST  DSNDB06       0 BP8K0  R N A N N      9         144      4   Y
SYSJAUXA DSNDB06       0 BP0    L N A N N      1         288      0 O Y
SYSJAUXB DSNDB06       0 BP0    L N A N N      1        1008      0 O Y
```

Figure 130. Example of a detailed report section - table spaces

Generating reports

Generating a report allows you to save or print information about Db2 objects for later use.

Procedure

To generate a report:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify any filtering criteria at the bottom of the panel and one of the following options, and press Enter:

   D
   Databases

   S
   Table Spaces

   T
   Tables, views, and aliases

   A
   Aliases

   Y
   Synonyms

   H
   Schemas

   E
   Data or distinct types

   F
   Functions

   G
   Storage groups

   O
   Stored procedures

J
: Triggers

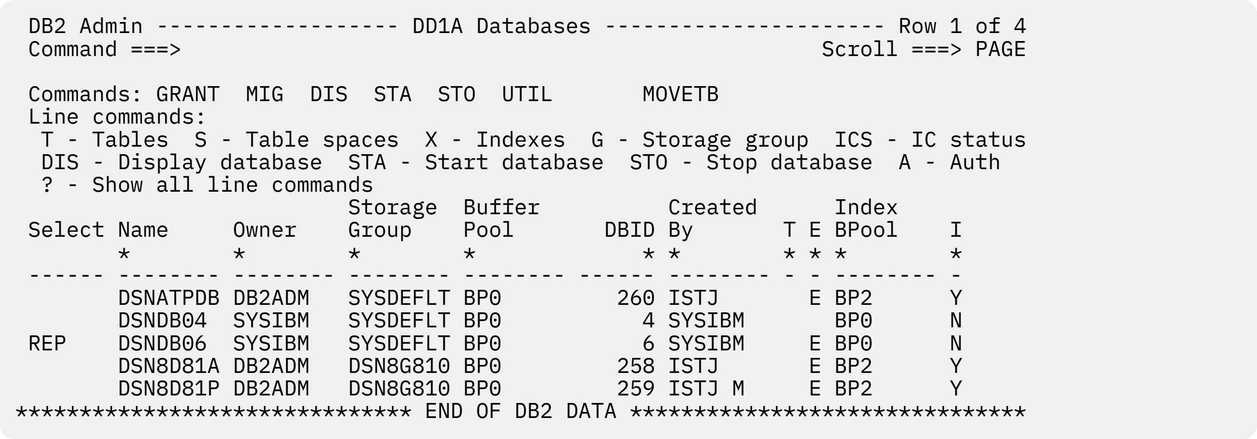
Q
: Sequences

These options support the ability to generate a report. You cannot generate reports for the other objects that are listed on the **System Catalog (ADB21)** panel.

3. On the resulting object panel, issue the REP line command or REP primary command as follows:

   - To generate a report for the single object, issue the REP line command.
   - To generate a report for all of the listed objects, issue the REP primary command.

   The following figure shows the REP command issued against a database.



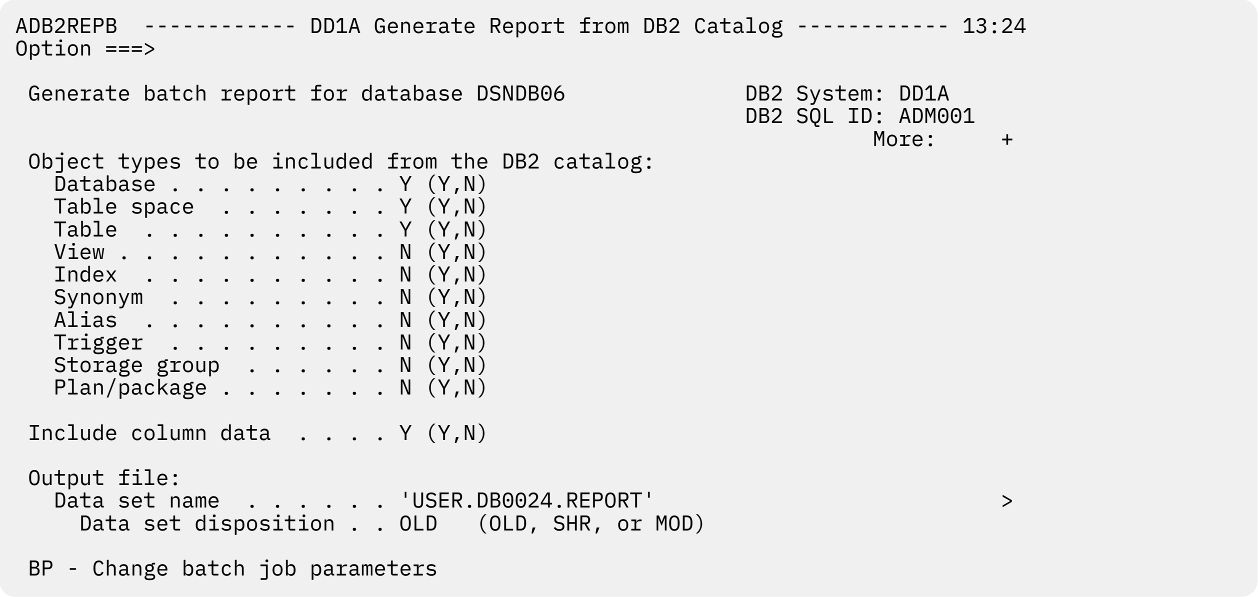
```
 DB2 Admin ------------------- DD1A Databases ---------------------- Row 1 of 4
 Command ===>                                                  Scroll ===> PAGE

 Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
 Line commands:
  T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
  DIS - Display database  STA - Start database  STO - Stop database  A - Auth
  ? - Show all line commands
                          Storage  Buffer          Created      Index
 Select Name     Owner    Group    Pool       DBID By       T E BPool    I
        *        *        *        *             * *        * * *        *
 ------ -------- -------- -------- -------- ------ -------- - - -------- -
        DSNATPDB DB2ADM   SYSDEFLT BP0         260 ISTJ       E BP2      Y
        DSNDB04  SYSIBM   SYSDEFLT BP0           4 SYSIBM       BP0      N
 REP    DSNDB06  SYSIBM   SYSDEFLT BP0           6 SYSIBM     E BP0      N
        DSN8D81A DB2ADM   DSN8G810 BP0         258 ISTJ       E BP2      Y
        DSN8D81P DB2ADM   DSN8G810 BP0         259 ISTJ M     E BP2      Y
******************************* END OF DB2 DATA *******************************
```

Figure 131. Databases panel (ADB21D) - Example of issuing the REP command to generate a report

4. On the **Generate Report from DB2 Catalog (ADB2REPB)** panel, specify the following items, and press Enter:

   - The types objects that you want included in the report. Specify Y for each object type that you want included in the report.
   - Whether to have the column properties for objects that have associated columns included in the report. This field is displayed only when it is applicable.
   - The data set information for the report output.

   The exact fields that are included on the panel depend on the type of object for which the REP command was issued.



```
ADB2REPB  ------------ DD1A Generate Report from DB2 Catalog ------------ 13:24
Option ===>

 Generate batch report for database DSNDB06             DB2 System: DD1A
                                                        DB2 SQL ID: ADM001
                                                                  More:     +
 Object types to be included from the DB2 catalog:
   Database . . . . . . . . . Y (Y,N)
   Table space  . . . . . . . Y (Y,N)
   Table  . . . . . . . . . . Y (Y,N)
   View . . . . . . . . . . . N (Y,N)
   Index  . . . . . . . . . . N (Y,N)
   Synonym  . . . . . . . . . N (Y,N)
   Alias  . . . . . . . . . . N (Y,N)
   Trigger  . . . . . . . . . N (Y,N)
   Storage group  . . . . . . N (Y,N)
   Plan/package . . . . . . . N (Y,N)

 Include column data  . . . . Y (Y,N)

 Output file:
   Data set name  . . . . . . 'USER.DB0024.REPORT'                          >
     Data set disposition . . OLD   (OLD, SHR, or MOD)

 BP - Change batch job parameters
```

Figure 132. ***Generate Report from DB2 Catalog (ADB2REPB)*** *panel*

The batch jobs to create the report are generated, and an ISPF Edit session is displayed.

5. Verify and submit the generated jobs.

   The report is created in a printed format and written to the data set that was specified.

Results

You are now ready to print the data set with the carriage control and specified rotate options.

Display of Db2 catalog columns

In some cases, Db2 Admin Tool displays the values of Db2 catalog columns differently than Db2 to improve readability.

Db2 Admin Tool has the following differences in displaying catalog column values:

- If a catalog column value is null, Db2 Admin Tool displays a question mark character (?).
- Db2 Admin Tool handles pairs of INTEGER and FLOAT columns differently than Db2 does.

  In some Db2 catalog tables, when a column with an INTEGER data type became too small to hold large values, Db2 added a corresponding column with a FLOAT data type to the catalog table to replace the INTEGER column. For example, CARDF was added for CARD in SYSTABLES, and FIRSTKEYCARDF was added for FIRSTKEYCARD in SYSINDEX. Db2 catalog tables (Db2 12 for z/OS) shows that the INTEGER version of the column is no longer used. When the catalog table is queried by a SELECT* statement in SPUFI , the value for the unused column might be displayed as 0 or -1.

  For readability, Db2 Admin Tool displays in the INTEGER column the integer equivalent of the value that is in the FLOAT column if the value fits. If the value is too large, Db2 Admin Tool displays 11 asterisks instead.

  The following columns contain the corresponding integer value of the floating point column when the floating point column is also included in the SELECT list:

  - CARD
  - COLCARD
  - FIRSTKEYCARD
  - FULLKEYCARD
  - FREQUENCY
  - FAROFFPOS

- NEAROFFPOS
- NACTIVE
- NPAGES
- SPACE
- KEYCOUNT
- CLUSTERRATIO

These integer columns must be returned by Db2 as INTEGER NOT NULL.

Within a SELECT statement in Db2 Admin Tool, use one of the following methods to get the real value from the Db2 catalog:

- Do not specify both the integer and floating point column in the SELECT list.
- Rename one of the columns in the SELECT list by using AS. For example: `SELECT CARD AS MYCARD`.
- Change the data type in the result. For example: `SELECT DECIMAL(COLCARD,11,0)`

Restrictions on Db2 object names

Db2 Admin Tool puts the following restrictions on Db2 object names.

Restrictions:

- Do not use object names that contain Unicode characters that cannot be translated into the EBCDIC CCSID that Db2 Admin Tool is using.
- Do not use object names that contain an apostrophe (').

Object names that contain these characters can be displayed, but when a line command is used with either of the restricted object names, an error (SQLCODE -104) or warning message might be displayed.

Masks

A *mask* provides the ability to make context-sensitive changes to naming conventions, overwrite the current values of certain object attributes, and verify that object attributes conform to certain rules.

You can use masks when you perform the following actions in Db2 Administration Tool:

- Generate SQL to reverse engineer Db2 objects (by using the GEN command).
- Clone a work statement list (WSL).
- Migrate Db2 object definitions, the data in those objects, or catalog statistics to other Db2 subsystems.
- Import changes through the Change Management process.

For example, when you migrate Db2 objects from one subsystem to another, you can use masks to change the name or owner of the objects on the target system. For instance, you can change all table space names to start with DB1A instead of DB1B.

You can also use masks when using IBM Db2 Object Comparison Tool for z/OS. In this case, a mask can be used for translation during the comparison. 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.

Depending on the function, masks are categorized as one of the following types:

Name masks
Change object names or qualifiers. Name masks can be also called *translation masks*, because they translate the names.

Overwrite masks
Overwrite the values of certain object attributes, such as COMPRESS, DEFINE, DEFER, and SEGSIZE.

Verification masks

Verify that object attributes conform to the specified rules, such as the PCTFREE attribute being greater than a certain amount. Verification masks are also called *VER masks*.

Name and overwrite masks can be used when running GEN, cloning a WSL, migrating objects, importing changes, and running Object Comparison Tool. Verification masks can be used only when running GEN or Object Comparison Tool.

You can define a mask either in a data set or in a Db2 table in the Change Management repository. If you define masks in a table, the Change Management repository tables must be defined as part of the customization process of Db2 Administration Tool. Masks that are specified when you import changes through Change Management must be defined in a table in the Change Management repository, not in a data set.

Recommendation: Consider managing all of your masks through Change Management. When masks are stored in a table in the Change Management database, they are easy to share, manage and recover.

Related information

Video: Generating DDL for multiple objects and using masks to change names

Mask definitions

Mask definitions are reusable; you can define a mask once and use it repeatedly.

Mask names, syntax, hierarchy, processing, and performance is the same whether you specify mask definitions in a data set or in the Change Management database.

Recommendation: Consider having an experienced DBA set up the masks initially. Other DBAs can then reuse the masks. For more recommendations, see "Mask recommendations" on page 305.

Syntax of mask definitions

The syntax of mask definitions depends on the type of mask you are creating:

Name masks

Use the following syntax to define a mask that changes a naming convention:

```
maskname: inputmask,outputmask
```

maskname

The name of the mask. Possible values are listed in Table 15 on page 275.

inputmask

The input mask, which is the string pattern that you want to translate. The specified input mask cannot be longer than 256 bytes.

outputmask

The output mask, which is the string to which you want to translate the input mask. The specified output mask cannot be longer than 256 bytes.

A SINGLECH mask is a type of name mask; it masks a single character and is used in conjunction with another name mask. Use the following syntax for a SINGLECH mask:

```
SINGLECH:character[,escape character]
maskname: inputmask,outputmask
```

character

A single character to use in the subsequent mask definition to indicate any character.

escape character

A character to be used as an escape character in the subsequent mask definition. If the specified escape character precedes *character* in the mask definition, *character* is treated as a literal.

Specifying an escape character is optional.

maskname*:*inputmask*,*outputmask

This line defines the name mask to which you want to apply the single character mask. Include *character* in *inputmask* or both *inputmask* and *outputmask* to indicate any single character in a particular position.

For examples of name masks and the SINGLECH mask, see "Examples of name mask definitions" on page 281.

Overwrite masks

Use the following syntax to define a mask that overwrites the value of a table space or index space attribute:

```
maskname: inputmask, overwrite_value
```

maskname

The name of the mask. For overwrite masks, the mask name is the name of the attribute that you want to overwrite. Possible values are listed in Table 16 on page 278.

inputmask

The table space name or index space name. The specified input mask cannot be longer than 256 bytes.

overwrite_value

The new value to use for the attribute. The value that you can specify depends on the attribute. For possible values, see Table 16 on page 278. The value can be a a string value such as YES or NO, an integer value (n), or an integer percentage of the current value (n%).

The value can also be a REXX user exit that calculates a value. See "Specifying a REXX user exit for a mask overwrite value" on page 304.

The specified overwrite value cannot be longer than 256 bytes. This length restriction also applies to the specification of a REXX user exit and its input variables.

For examples of overwrite masks, see "Examples of overwrite mask definitions" on page 282.

Verification masks

For syntax and examples, see "Verification masks" on page 289.

Masks can contain generic specifications, which are expressed by using an asterisk (*).

Use a plus sign (+) in column 72 to indicate the continuation of a mask definition onto the next line. The plus sign (+) can also be used in the middle of specifying a REXX user exit in the mask definition to indicate that the specification continues on the next line.

You can specify one or more masks. For information about how multiple mask definitions are processed, see "Mask order" on page 286.

Mask names

For name masks, which change a naming convention, valid mask names are listed in Table 15 on page 275.

For overwrite masks, which overwrite object attributes, valid mask names are listed in Table 16 on page 278.

Table 15. Masks for translating names

Name	Description
SINGLECH	Single characters
COLNAME	Column names
NAME	All object names (This mask affects all masks that have NAME as a parent or grandparent. See the mask hierarchy in Table 17 on page 283.)

Table 15. Masks for translating names (continued)

Name	Description
BPNAME	All buffer pool names
IXBPNAME	Buffer pool names for indexes
TSBPNAME	Buffer pool names for table spaces
COLLNAME	Collection names
CONSNAME	Constraint names
DBNAME	Database names
DBRMNAME [1]	DBRM names (Used for BIND commands.)
GBPNAME [1]	Group buffer pool names
GRPNAME	Group names
GVNAME	Names of global variables
IXNAME	Index names
MKNAME	Names of column masks
PGMNAME	Program names; synonym for DBRM names
PKGNAME	Package names
PLNNAME	Plan names
PMNAME	Names of row permissions
SEQNAME	Sequence names
SFNAME [1]	Specific function names
SGNAME	All storage group names
IXSGNAME	Storage group names for indexes
TSSGNAME	Storage group names for table spaces
STPNAME	Stored procedure names and specific names for native SQL procedures
TBNAME	Table, alias, synonym, and view names
ALNAME	Alias names (This mask is valid only for those CREATE statements where the object is clearly defined as an alias.)
SYNNAME	Synonym names
VWNAME	View names (This mask is valid only for those CREATE statements where the object is clearly defined as an view.)
TCNAME	Trusted context names
TGNAME	Trigger names
TSNAME	Table space names
UDFNAME	User-defined function names
UDTNAME	User-defined data type names
VCATNAME	VCAT names
GRANTEE	Grantees

Table 15. Masks for translating names (continued)

Name	**Description**
GRANTOR	Grantors
OWNER [2, 3]	Owners and creators (This mask affects all masks that have OWNER as a parent. See the mask hierarchy in Table 17 on page 283.)
DBOWNER [2, 3]	Database owners
IXOWNER [2, 3]	Index owners (IXOWNER masks the index creator field, which is the owner of the index in DB2 UDB for z/OS 8, but is the schema of the index in DB2 9 for z/OS.)
PKGOWNER [2, 3]	Package owners
SGOWNER [2, 3]	Storage group owners
TBOWNER [2, 3]	Table owners (TBOWNER masks the table creator field, which is the owner of the table in DB2 UDB for z/OS 8, but is the schema of the table in DB2 9 for z/OS.)
SYNOWNER (subset of TBOWNER) [2, 3]	Synonym owners
TSOWNER [2, 3]	Table space owners
SCHEMA [3]	Schemas (This mask affects all masks that have SCHEMA as a parent. See the mask hierarchy in Table 17 on page 283.)
GVSCHEMA [3]	Global variable schemas
IXSCHEMA [3]	Index schemas (IXSCHEMA masks the index creator field, which is the owner of the index in DB2 UDB for z/OS 8, but the SCHEMA of the index in DB2 9 for z/OS or later.)
MKSCHEMA [3]	Column mask schemas
PMSCHEMA [3]	Row schemas
SEQSCHEMA [3]	Sequence schemas
SETPATHSC	SET CURRENT PATH schemas
STPSCHEMA	Stored procedure schemas
TBSCHEMA	Table schemas (TBSCHEMA masks the table creator field, which is the owner of the table in DB2 UDB for z/OS 8, but the schema of the table in DB2 9 for z/OS or later.)
ALSCHEMA (subset of TBSCHEMA)	Alias schemas (This mask is valid only for those CREATE statements where the object is clearly defined as an alias.)
VWSCHEMA (subset of TBSCHEMA)	View schemas (This mask is valid only for those CREATE statements where the object is clearly defined as an view.)
TGSCHEMA	Trigger schemas
UDFSCHEMA	Function schemas
UDTSCHEMA	User-defined type schemas
ROLE	Role names
DBROLE [4]	Roles that are associated with a database

Table 15. Masks for translating names (continued)

Name	Description
IXROLE [4]	Roles that are associated with an index
TBROLE[4]	Roles that are associated with a table
TSROLE [4]	Roles that are associated with a table space
SQLID[1]	Current SQLIDs This mask is needed by cloning when masking SET CURRENT SQLID statements that are already generated. When the GEN function and migrate function generate SET CURRENT SQLID = *sqlid* statements, the SQLID mask is not used to mask the *sqlid* value. The *sqlid* in these statements originates from field values in the Db2 catalog, and these values are masked before the SET statement is generated. For example, CREATE SYNONYM requires a SET CURRENT SQLID statement to set the current SQLID to the synonym owner (creator). The OWNER mask is used to mask the synonym owner before the SET statement is generated.
XMLSCHID	Registered XML schema names in an XML-type modifier
WLMENV	Workload Manager (WLM) environment names
LOCATION	Location names, where the location is the first of a three-part name, as in: *location.schema.name*

Notes:

1. DBRMNAME, GBPNAME, SFNAME, and SQLID are used only when work statement lists (WSLs) are cloned. If specified, they have no affect on the GEN function, the migrate function, or importing changes.
2. All OWNER masks have no affect when WSLs are cloned and when importing changes. (This condition applies to the OWNER mask and all masks that have OWNER as a parent. See the mask hierarchy in Table 17 on page 283.)
3. 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. For example:

   ```
   SCHEMA:maskA;maskB
   OWNER:maskA;maskB
   ```

   For objects that were created in DB2 9 or an earlier version of Db2, the schema and owner had the same value. However, both Db2 Admin Tool and Object Comparison Tool treat schema and owner as distinct values and do not assume that the values are the same.
4. The DBROLE, TSROLE, TBROLE, and IXROLE masks are not currently used.

Table 16. Masks for overwriting attributes

Name	Description	Possible values[1, 2, 3]
APPEND	Whether the APPEND option is specified for the table	YES, NO
AUDIT	AUDIT option for a table	CHANGES, ALL, NONE
CLOSE	Whether the table space or index space data set is eligible to be closed	YES, NO
TSCLOSE	Whether the table space data set is eligible to be closed	YES, NO

Table 16. Masks for overwriting attributes (continued)

Name	Description	Possible values[1, 2, 3]
IXCLOSE	Whether the index space data set is eligible to be closed	YES, NO
COMPRESS	Whether a table space or table space partition is compressed	YES, NO
TSCOMPRES	Whether a table space or table space partition is compressed	YES, NO, FIXED, HUFFMAN[5]
IXCOMPRES	Whether an index is compressed	YES, NO
COPY	Whether COPY YES was specified for the index	YES, NO
DCAPTURE	DATA CAPTURE option for a table	NONE, CHANGES
DEFER	Whether to build the index when the CREATE INDEX statement is executed	YES, NO
DEFINE	Whether the underlying data sets for the table space or index space are created when the object is created (NO indicates that the data sets are not created until data is inserted into the object.)	YES, NO
IXDEFINE	Whether the underlying data sets for the index space is created when the index space is created. (NO indicates that the data sets are not created until data is inserted into the object.)	YES, NO
TSDEFINE	Whether the underlying data sets for the table space are created when the table space is created (NO indicates that the data sets are not created until data is inserted into the object.)	YES, NO
DSSIZE	Maximum size, in gigabytes, for each partition in a partitioned table space	*n*, where *n* is a power of two, in the range 1 - 256
TSDSSIZE	DSSIZE attribute for table spaces	*n*G
IXDSSIZE	DSSIZE attribute for indexes	*n*G
DTINLOBL	INLINE LENGTH integer value for distinct types	*n*
TBINLOBL	INLINE LENGTH integer value for tables	*n*
EDITPROC	Name of the edit procedure that is associated with the table	*string*
ERASE	Whether the Db2-managed data sets are to be erased	YES, NO
TSERASE	Whether the Db2-managed data sets are to be erased for table spaces	YES, NO
IXERASE	Whether the Db2-managed data sets are to be erased for indexes	YES, NO
FIELDPROC	The name of the field procedure that is associated with a column	*string*

Table 16. Masks for overwriting attributes (continued)

Name	Description	Possible values[1, 2, 3]
FREEPG	Number of pages that are loaded before a page is left as free space	*n*
TSFREEPG	Number of pages that are loaded before a page is left as free space for table spaces	*n*
IXFREEPG	Number of pages that are loaded before a page is left as free space for indexes	*n*
HASHSPC	HASH SPACE integer	*n*K, *n*M, *n*G
INSALGO	INSERT ALGORITHM attribute of a table space	0, 1, or 2
IXGBPCACH	Group buffer pool cache option for the index or index partition	SYSTEM, CHANGED, ALL, NONE
TSGBPCACH	Group buffer pool cache option for the table space or table space partition	SYSTEM, CHANGED, ALL, NONE
LOCKSIZE	Lock size of the table space	TABLE, TABLESPACE, PAGE, ROW, LOB, ANY
LOCKMAX	Maximum number of locks that need to be acquired per user for the table or table space before escalating to the next locking level	*n*, SYSTEM
LOGGED	Whether the changes to a table space are logged	YES, NO
MAXROWS	Maximum number of rows that Db2 is to place on a data page	*n*
MEMCLUS	Whether or not MEMBER CLUSTER is specified for the table space	YES, NO
PADDED	Whether keys within the index are padded for varying-length column data	YES, NO
PCTFREE	Percentage of each page that is left as free space	*n*
TSPCTFREE	Percentage of each page that is left as free space for table spaces	*n*
TSPCTFUPD	Percentage of free space that is reserved for updates to variable length records, as defined when the table space is created or altered	*n*
IXPCTFREE	Percentage of each page that is left as free space for indexes	*n*
PRIQTY	Minimum primary space allocation for a Db2-managed data set for table spaces and index spaces	*n*, *n*%
IXPRIQTY	Minimum primary space allocation for a Db2-managed data set for index spaces	*n*, *n*%
TSPRIQTY	Minimum primary space allocation for a Db2-managed data set for table spaces	*n*, *n*%
RESONDROP	RESTRICT ON DROP attribute for tables	YES, NO

Table 16. Masks for overwriting attributes (continued)

Name	Description	Possible values[1, 2, 3]
SECQTY	Minimum secondary space allocation for a Db2-managed data set for table spaces and index spaces	*n*, *n*%
IXSECQTY	Minimum secondary space allocation for a Db2-managed data set for index spaces	*n*, *n*%
TSSECQTY	Minimum secondary space allocation for a Db2-managed data set for table spaces	*n*, *n*%
SEGSIZE	Number of pages in each segment of a segmented table space	*n*, where *n* is a multiple of 4, in the range of 4 - 64
SGKEYLABL[4]	The key label for the storage group	*string*, NO, NOKEYLABEL
TBKEYLABL[4]	The key label for the table	*string*, NO, NOKEYLABEL
TRACKMOD	Whether to track page modifications in the space map	YES, NO
TSPARTS	Number of partitions of the table space	*n* (0 if the table space is not partitioned)
VALIDPROC	Name of the validation procedure that is associated with the table	*string*
VOLATILE	Whether Db2 uses index access to the table whenever possible for SQL operations	YES, NO

Notes:

1. The values for these overwrite masks can also be a REXX user exit that returns a valid value for the given mask. See "Specifying a REXX user exit for a mask overwrite value" on page 304.
2. *n* = an integer
3. *n%* = an integer percentage of the current value
4. These overwrite masks apply to only Db2 12 function level 502 or higher.
5. The FIXED and HUFFMAN values for TSCOMPRES apply to only Db2 12 function level 509 or higher.

Examples of name mask definitions

NAME mask example

The following mask changes any name that starts with ABC to a name that starts with DEF in the generated SQL.

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

AUTHID mask example

The following mask translates all authorization IDs that have the value SYSIBM to COPY.

```
AUTHID: SYSIBM, COPY
```

TBNAME mask example

The following mask can be used to translate a table that is named EMPLOYEE01 to EMPLOYEE02.

```
TBNAME: *01*, *02*
```

COLNAME mask example

The following mask changes any column name that starts with COL in any table. In this case, the column name is changed to start with NEWCOL. The column names that are changed include column names in triggers, views, and indexes. You cannot selectively change column names in specific tables.

```
COLNAME: COL*, NEWCOL*
```

SINGLECH mask examples

The following masks change any character in position 2 of the matching database names to a 9.

```
SINGLECH:#
DBNAME:D#J12345, D9J12345
```

With the following masks, the table name ABC is translated to DBF:

```
SINGLECH:_
TBNAME:A_C,D_F
```

The following masks specify that for any name that matches the input mask, any character in the seventh position is to be replaced with FL. For example, ADMIN_TASKS is to be replaced with ADMIN_FLASKS. Notice that the plus sign (+) is defined as an escape character; when it used in the name mask, the following underscore (_) is treated as a literal.

```
SINGLECH:_,+
NAME:ADMIN+__*,ADMIN+_FL*
```

Examples of overwrite mask definitions

COMPRESS attribute mask example

The following mask changes any table spaces that start with TESTTS in the TESTDB database to be compressed.

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

PRIQTY attribute mask example

The following mask changes the PRIQTY value of all table spaces and index spaces in TESTDB database to 75% of the current value of PRIQTY.

```
PRIQTY: TESTDB.*, 75%
```

Example of specifying a REXX user exit for an overwrite value

The following mask changes table spaces that start with TESTTS in the TESTDB database to use the DSSIZE value that is returned by the REXX user exit PDDSSIZE.

```
DSSIZE: TESTDB.TESTTS*, REXX(PDDSSIZE,PARTITIONS,BPOOL)
```

Mask Hierarchy

Mask specifications can use high-level options, such as NAME, AUTHID, SCHEMA, OWNER, GRANTID, or ROLE, or more granular options such as DBNAME, TSNAME, SQLID, TBSCHEMA, DBOWNER, or GRANTOR. This defined hierarchy of masks is shown in the following tables.

The following table shows the hierarchy of name masks. Name masks that are not listed in this table do not participate in a hierarchy. For example, the COLNAME mask has no levels and does not participate in a hierarchy. To translate a column name, you must use COLNAME.

Table 17. Hierarchy of name masks

Grandparent	Parent	Child
NAME	BPNAME	IXBPNAME
		TSBPNAME
	COLLNAME	
	CONSNAME	
	DBNAME	
	DBRMNAME	
	GBPNAME	
	GRPNAME	
	GVNAME	
	IXNAME	
	MKNAME	
	PGMNAME	
	PKGNAME	
	PLNNAME	
	PMNAME	
	SEQNAME	
	SFNAME	
	SGNAME	IXSGNAME
		TSSGNAME
	STPNAME	
	TBNAME	
	TBNAME	ALNAME
		SYNNAME
		VWNAME
	TCNAME	
	TGNAME	
	TSNAME	
	UDFNAME	
	UDTNAME	
	VCATNAME	

Table 17. Hierarchy of name masks (continued)

Grandparent	Parent	Child
AUTHID	GRANTID	GRANTEE
		GRANTOR
	OWNER	DBOWNER
		IXOWNER
		PKGOWNER
		SGOWNER
		TBOWNER
		SYNOWNER (subset of TBOWNER)
		TSOWNER
	SCHEMA	GVSCHEMA
		IXSCHEMA
		MKSCHEMA
		PMSCHEMA
		SEQSCHEMA
		SETPATHSC
		STPSCHEMA
		TBSCHEMA
		ALSCHEMA (subset of TBSCHEMA)
		VWSCHEMA (subset of TBSCHEMA)
		TGSCHEMA
		UDFSCHEMA
		UDTSCHEMA
	ROLE	DBROLE
		IXROLE
		TBROLE
		TSROLE
	SQLID	

The following table shows the hierarchy of overwrite masks. Overwrite masks that are not listed in this table do not participate in a hierarchy.

Table 18. Hierarchy of overwrite masks

Parent	Child
CLOSE	TSCLOSE
	IXCLOSE

Table 18. Hierarchy of overwrite masks (continued)

Parent	Child
COMPRESS	TSCOMPRES
	IXCOMPRES
DEFINE	IXDEFINE
	TSDEFINE
ERASE	TSERASE
	IXERASE
FREEPG	IXFREEPG
	TSFREEPG
GBPCACH	IXGBPCACH
	TSGBPCACH
PCTFREE	IXPCTFREE
	TSPCTFREE
PRIQTY	IXPRIQTY
	TSPRIQTY
SECQTY	IXSECQTY
	TSSECQTY
DSSIZE	IXDSSIZE
	TSDSSIZE

If you use a mask data set to edit or view mask definitions, you can also see a list of mask names and their hierarchy in the MSG lines of the mask data set. See "Creating masks in a data set" on page 309.

Examples of mask heirarchy

DBNAME example

To translate a database name so that it starts with Y instead of X, you can specify either of the following masks:

```
DBNAME: X*,Y*
```

```
NAME: X*,Y*
```

The NAME mask is a grandparent in the hierarchy and, therefore, more general than the DBNAME mask, which is a child of NAME. Using the higher-level NAME mask affects all masks that are children or grandchildren of NAME.

BPNAME example

The BPNAME mask (for buffer pool name) has three levels: TSBPNAME, BPNAME and NAME. To translate a table space buffer pool name, you can use either TSBPNAME, BPNAME, or NAME. Depending on the situation, the NAME mask is probably too general. However, even the BPNAME mask might be too general. The BPNAME mask affects the matching names for all table space buffer pools and index space buffer pools. The lower-level TSBPNAME affects only the names of table space buffer pools.

TSPRIQTY example

TSPRIQTY is second in the hierarchy of PRIQTY and TSPRIQTY. TSPRIQTY overwrites the PRIQTY for table spaces only, whereas PRIQTY overwrites the PRIQTY for both table spaces and index spaces.

Mask order

You can specify as many masks as you want. Masks are processed in the order that you list them. The first mask that matches is used. 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 the *inputmask* in the mask syntax. For example, a table named PRODTAB1 conforms to input 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 masks first and the more general ones at the end.

Performance tip: Using many masks might increase processing time. If a match is not found early in the process, the program must search through the list of masks until a match is found.

Object-specific masks

The effects of some naming masks are too general for all situations. For example, the IXBPNAME mask changes the name of every matching instance of an index buffer pool. If you need to change a buffer pool name for only one index, you can use specify that a mask applies to only a specific object. This type of mask is called an *object-specific mask*. Object-specific masks change only names, not attributes.

Use the following syntax to define an object-specific mask:

```
maskname:qual.name:inputmask,outputmask
```

maskname

The name of the mask. Possible values are listed in "Mask names" on page 275

qual.name

The name of the object. A qualifier (*qual*) is optional.

name does not always refer to the name of the masked item. For example, for the IXSGNAME mask, *name* refers to the index name, not the storage group name. To determine which object needs to be specified, see Table 19 on page 286.

inputmask

The input mask, which is the string pattern that you want to change. When you use object-specific masking, the input mask can be greater than 256 bytes.

outputmask

The output mask, which is the string to which you want to translate the input mask. The maximum length allowed for an output mask is 256 bytes.

The following tables lists all of the object-specific masks.

Table 19. Object-specific masks and the objects they affect

Name	Syntax
ALNAME	ALNAME:*alias_schema.alias_name*:*current_alname,new_alname*
ALSCHEMA	ALSCHEMA:*alias_schema.alias_name*:*current_alschema,new_alschema*
COLNAME	COLNAME:*table_schema.table_name*:*current_colname,new_colname*
CONSNAME	CONSNAME:*table_schema.table_name*:*current_consname,new_consname*
DBOWNER	DBOWNER:*database_name*:*current_dbowner,new_dbowner*

Table 19. Object-specific masks and the objects they affect (continued)

Name	**Syntax**
DBRMNAME[1]	DBRMNAME:*stp_schema.stp_name*:*current_dbrmname,new_dbrmname*
DBRMNAME	DBRMNAME:*udf_schema.udf_name*:*current_dbrmname,new_dbrmname*
DBRMNAME	DBRMNAME:*table_schema.table_name*:*current_dbrmname,new_dbrmname*
GRPNAME	GRPNAME:*database_name*:*current_grpname,new_grpname*
GVNAME	GVNAME:*gv_schema.gv_name*:*current_gvname,new_gvname*
GVSCHEMA	GVSCHEMA:*gv_schema.gv_name*:*current_gvschema,new_gvschema*
IXBPNAME[2]	IXBPNAME:*index_schema.index_name*:*current_bpname,new_bpname*
IXBPNAME	IXBPNAME:*database_name*:*current_db_indexbpname,new_db_indexbpname*
IXNAME	IXNAME:*index_schema,index_name*:*current_ixname,new_ixname*
IXOWNER	IXOWNER*:index_schema.index_name*:*current_ixowner,new_ixowner*
IXSCHEMA	IXSCHEMA:*index_schema.index_name*:*current_ixschema,new_ixschema*
IXSGNAME	IXSGNAME:*index_schema.index_name*:*current_ixsgname,new_ixsgname*
LOCATION	LOCATION:*schema_name.obj_name*:*current_location,new_location*
MKNAME	MKNAME:*mask_schema.mask_name*:*current_maskname,new_maskname*
MKSCHEMA	MKSCHEMA:*mask_schema.mask_name*:*current_mkschema,new_mkschema*
PGMNAME[1]	PGMNAME:*stp_schema.stp_name*:*current_pgmname,new_pgmname*
PGMNAME	PGMNAME:*udf_schema.udf_name*:*current_pgmname,new_pgmname*
PGMNAME	PGMNAME:*table_schema.table_name*:*current_pgmname,new_pgmname*
PKGNAME	PKGNAME:*collection_id.package_name*:*current_pkgname,new_pkgname*
PKGOWNER	PKGOWNER:*collection_id.package_name*:*current_packageowner,new_packageowner*
PMNAME	PMNAME:*pm_schema.pm_name*:*current_pmname,new_pmname*
PMSCHEMA	PMSCHEMA:*pm_schema.pm_name*:*current_pmschema,new_pmschema*
SEQNAME	SEQNAME:*seq_schema.seq_name*:*current_seqname,new_seqname*
SEQSCHEMA	SEQSCHEMA:*seq_schema.seq_name*:*current_seqschema,new_seqschema*
SGOWNER	SGOWNER:*stogroup_name*:*current_stogroupowne*r, *new_stogroupowner*
STPNAME	STPNAME:*stp_schema.stp_name*:*current_stpname,new_stpname*
STPSCHEMA	STPSCHEMA:*stp_schema.stp_name*:*current_stpschema,new_stpschema*
SYNNAME	SYNNAME:*synonym_owner.synonym_name*:*current_syname,new_syname*
SYNOWNER	SYNOWNER:*synonym_owner.synonym_name*:*current_synowner,new_synowner*
TBNAME	TBNAME:*table_schema.table_name*:*current_tbname,new_tbname*
TBOWNER	TBOWNER:*table_schema.table_name.current_tbowner,new_tbowner*
TBSCHEMA	TBSCHEMA:*table_schema.table_name*:*current_tbschema,new_tbschema*
TGNAME	TGNAME:*trigger_schema.trigger_name*:*current_tgname,new_tgname*
TGSCHEMA	TGSCHEMA:*trigger_schema.trigger_name*:*current_tbschema,new_tgschema*
TSBPNAME[2]	TSBPNAME:*database_name.tablespace_name*:*current_tspbname,new_tsbpname*
TSBPNAME	TSBPNAME:*database_name*:*current_dbbpname,new_dbbpname*
TSNAME	TSNAME:*database_name.tablespace_name*:*current_tsname,new_tsname*
TSOWNER	TSOWNER:*database_name.tablespace_name*:*current_tsowner,new_tsowner*
TSSGNAME[2]	TSSGNAME:*database_name.tablespace_name*:*current_tssgname,new_tssgname*

Table 19. Object-specific masks and the objects they affect (continued)

Name	Syntax
TSSGNAME	TSSGNAME:*database_name*:*current_dbsgname*,*new_dbsgname*
UDFNAME	UDFNAME:*udf_schema.udf_name*:*current_udfname*,*new_udfname*
UDFSCHEMA	UDFSCHEMA:*udf_schema.udf_name*:*current_udfschema*,*new_udfschema*
UDTNAME	UDTNAME:*udt_schema.udt_name*:*current_udtname*,*new_udtname*
UDTSCHEMA	UDTSCHEMA:*udt_schema.udt_name*:*current_udtschema*,*new_udtschema*
VCATNAME[1]	VCATNAME:*stogroup_name*:*current_vcatname*, *new_vcatname*
VCATNAME	VCATNAME:*schema.obj_name*:*current_vcatname*,*new_vcatname*
VWNAME	VWNAME:*view_schema.view_name*:*current_vwname*,*new_vwname*
VWSCHEMA	VWSCHEMA:*view_schema.view_name*:*current_vwschema*,*new_vwschema*
WLMENV	WLMENV:*udf_schema.udf_name*:*current_wlmenvname.new_wlmenvname*
WLMENV	WLMENV:*stp_schema.stp_name*:*current_wlmenvname*,*new_wlmenvname*

Notes:

1. The DBRMNAME, PGMNAME, and VCATNAME masks can be used for more than one object type.
2. The IXBPNAME, TSPBNAME, and TSSGNAME masks can be used for both object-level and database-level versions of the names.

The following masks cannot have object-specific qualifiers:

- SINGLECH
- NAME
- BPNAME
- COLLNAME
- DBNAME
- GBPNAME
- PLNNAME
- SFNAME
- SGNAME
- TCNAME
- AUTHID
- GRANTID
- GRANTEE
- GRANTOR
- OWNER
- SCHEMA
- SETPATHSC
- ROLE
- DBROLE
- IXROLE
- TBROLE
- TSROLE
- SQLID
- XMLSCHID

Examples of object-specific masks

TBNAME object-specific mask example

The following mask applies to only the CREATOR1.TB2 table:

```
TBNAME:CREATOR1.TB2:CREATOR1,NEW_CRE1
```

IXBPNAME object-specific mask example

The following mask changes the buffer pool name to BP3 for only index IXOWN1.IX2:

```
IXBPNAME:IXOWN1.IX2:BP1,BP3
```

TSSGNAME object-specific mask example

The following mask changes the storage group name for only database TESTDB from SG1 to SG0.

```
TSSGNAME: TESTDB : SG1,SG0
```

TSBPNAME object-specific mask example

The following mask changes the bufferpool name BP0 to BP1 for all table spaces that start with TESTTS in the TESTDB database.

```
TSBPNAME: TESTDB.TESTTS* : BP0,BP1
```

VWSCHEMA object-specific mask example

The following mask changes the view schema for all views that start with VWA and have schema names that start with SCH. In this case, TEST is added to the end of these schema names.

```
VWSCHEMA: SCH*.VWA* : *, *TEST
```

Verification masks

You can use verification masks, also called VER masks, to verify that Db2 object attributes conform to certain rules. The process of using these masks is also know as a *verification request*, or VER request.

Restriction: You can use verification masks only with the GEN function and IBM Db2 Object Comparison Tool for z/OS.

You can specify both verification masks and other masks. Verification masks are applied first, and then the other masks.

Syntax of verification masks

Use one of the following syntax formats for a verification mask:

- `VER,field:verification_operator,verification_values,RC=verification_rc`
- `VER,rexx-field:REXX(exec_name,parm_1parm_2,...parm_n)`

VER

Indicates that the mask is a verification mask.

field

The attribute to verify. *field* must be an overwrite mask name. For a list of valid overwrite mask names, see Table 16 on page 278. All overwrite masks are allowed for verification masks.

verification_operator

The operator to use for the verification. This operator is used to compare the specified attribute (*field*) with the value or values (*verification_values*).

verification_operator can have one of the following values:

EQ

Equal comparison

NE
Not equal comparison

LT
Less than comparison

GT
Greater than comparison

LIST
List of values

RANGE
Greater than or equal to the first value, and less than or equal to the second value.

verification_values
The value or values to compare with *field*.

verification_rc
The return code that is issued if the comparison expression is false.

verification_rc can have one of the following values:

12
Fatal. Processing fails.

8
Severe. Processing fails.

6
Error. Processing fails.

4
Warning. Processing fails only if no exceptions are granted.

0
Processing does not fail.

rexx-field
The attribute or object that you want to verify with a REXX exec. *rexx- field* can have one of the following values:

- An overwrite mask name. For a list of valid overwrite mask names, see Table 16 on page 278.
- OBJNAME

  OBJNAME is a special verification mask type that is allowed only with the REXX exec syntax. An OBJNAME verification mask can verify both the object name and the object schema. If you specify OBJNAME, you need to provide three arguments to the REXX exec: object type, object name and object schema.
- One of the following two-character object type codes:

Object type code	Object type	Corresponding Db2 catalog table
SG	Storage group	SYSSTOGROUP
DB	Database	SYSDATABASE
TS	Table space	SYSTABLESPACE
TB	Table	SYSTABLES
IX	Index	SYSINDEXES
TG	Trigger	SYSTRIGGERS
FK	Foreign Key	SYSRELS
PK	Primary key	SYSTABCONST

Object type code	Object type	Corresponding Db2 catalog table
CK [1]	Check Constraint	SYSCHECKS
UQ	Unique Constraint	SYSTABCONST
DT	Data type	SYSDATATYPES
FU	Function	SYSROUTINES
SP	Procedure	SYSROUTINES
SQ	Sequence	SYSSEQUENCES
SY	Synonyms	SYSSYNONYMS
AL	Alias	SYSTABLES
VW	View	SYSVIEWS
GV	Global variable	SYSVARIABLES
TP	Table part	SYSTABLEPART
IP	Index part	SYSINDEXPART
FL	Fields	SYSFIELDS

Notes:

1. The CK mask is not triggered for those check constraints that are generated by Db2, such as check constraints for business time period and system time period.

REXX
Indicates that a REXX exec is to be used to validate the field.

The REXX exec must return two items in the return string:

- The first two characters are the return code value, followed by a colon.
- If the REXX exec finds an error in verification, the rest of the string is the error message. Otherwise, the remaining string is blank.

exec_name
The name of the REXX exec.

parm_1,parm_2,parm_n
The parameters to pass to the REXX exec.

Example verification masks

The following VER mask verifies whether the COMPRESS attribute equals Y. If not, the mask issues return code 4.

```
VER,COMPRESS:EQ,Y,RC=4
```

The following VER mask verifies whether the PCTFREE attribute is greater than 20. If not, the mask issues return code 8.

```
VER,PCTFREE:GT,20,RC=8
```

The following VER mask verifies whether the PCTFREE attribute is within the range of 0-5, inclusive. If not, the mask issues return code 4.

```
VER,PCTFREE:RANGE,0,5,RC=4
```

The following VER mask limits the value of the table space attribute INSERT ALGORITHM to 0 or 1. If not, the mask issues return code 8.

```
VER,INSALGO:LIST,0,1,RC=8
```

Verification errors

If any verification errors are encountered, the calling function stops the process after all verification masks are applied. Verification errors are written to a separate data set called VALOUT, which uses SYSOUT. If you are in TSO mode, use the `TSO ISRDDN` command to see the VALOUT data set.

Related concepts

"Masks" on page 273
A *mask* provides the ability to make context-sensitive changes to naming conventions, overwrite the current values of certain object attributes, and verify that object attributes conform to certain rules.

Db2 catalog columns and the corresponding masks

Object names and attributes are stored in columns the Db2 catalog. If you know which catalog column information you want to mask, you can use the following table to determine which mask to use.

Table 20. Catalog columns and the corresponding masks

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSAUXRELS	AUXTBNAME	TBNAME	
	AUXTBOWNER	OWNER	
		TBOWNER	
	COLNAME	COLNAME	
	TBNAME	TBNAME	
	TBOWNER	OWNER	
		TBOWNER	
SYSCHECKS	CHECKCONDITION	COLNAME	Mask column names
	CREATOR	OWNER	
	TBNAME	TBNAME	
	TBOWNER	OWNER	
		TBOWNER	
SYSCOLAUTH	COLLID	COLLNAME	If grantee is package
	COLNAME	COLNAME	
	CREATOR	OWNER	
		TBOWNER	
	GRANTEE	PKGNAME	If grantee is package
		PLNNAME	If grantee is plan
		GRANTEE	If grantee is an authorization ID
	GRANTOR	GRANTOR	
	TNAME	TBNAME	

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSCOLUMNS	CHECKCONDITION	COLNAME	Mask column names
	CREATOR	OWNER	
	LENGTH	TBINLOBL	If Length is greater than 4 for INLINE LOB columns
	NAME	COLNAME	
	TBCREATOR	OWNER	If schema not SYSIBM
		TBOWNER	If schema not SYSIBM
	TBNAME	TBNAME	
	TBOWNER	OWNER	
		TBOWNER	
	TYPENAME	UDTNAME	If schema not SYSIBM
	TYPESCHEMA	SCHEMA	
SYSCONTROLS	NAME	MKNAME	
		PMNAME	
	SCHEMA	MKSCHEMA	If control_type is column mask
		PMSCHEMA	If control_type is row permission
SYSDATABASE	BPOOL	TSBPNAME	
	CREATOR	OWNER	
		DBOWNER	
	GROUP_MEMBER	GRPNAME	
	INDEXBP	IXBPNAME	
	NAME	DBNAME	
	STGROUP	TSSGNAME	
SYSDATATYPES	INLINE_LENGTH	DTINLOB	If distinct type is based on LOB source type
	NAME	UDTNAME	
	OWNER	OWNER	
	SCHEMA	SCHEMA	
SYSDBAUTH	GRANTEE	GRANTEE	
	GRANTOR	GRANTOR	
	NAME	DBNAME	
SYSFIELDS	FLDPROC	FIELDPROC	
		PGMNAME	
	NAME	COLNAME	
	TBCREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSFOREIGNKEYS	COLNAME	COLNAME	
	CREATOR	OWNER	
		TBOWNER	
	RELNAME	NAME	
	TBNAME	TBNAME	
SYSVARIABLES	NAME	GVNAME	
	OWNER	OWNER	
	SCHEMA	GVSCHEMA	
SYSVARIABLEAUTH	GRANTEE	GRANTEE	
	GRANTOR	GRANTOR	
	NAME	GVNAME	
	SCHEMA	GVSCHEMA	

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSINDEXES	BPOOL	IXBPNAME	
	CLOSERULE	CLOSE	
		IXCLOSE	
	COMPRESS	IXCOMPRES	
	COPY	COPY	
	CREATOR	OWNER	
		IXOWNER	
	DBNAME	DBNAME	
	DSSIZE	DSSIZE	
		IXDSSIZE	
	ERASERULE	ERASE	
		IXERASE	
	FREEPAGE	FREEPG	
		IXFREEPG	
	GBPCACHE	GBPCACHE	
		IXGBPCACH	
	NAME	IXNAME	
	PADDED	PADDED	
	PCTFREE	PCTFREE	
		IXPCTFREE	
	PQTY	PRIQTY	
		IXPRIQTY	
	SECQTYI	SECQTY	
		IXSECQTY	
	STORNAME	SGNAME	
		IXSGNAME	
	STORTYPE	SGTYPE	
		IXSGTYPE	
	TBCREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	
	VCATNAME	VCATNAME	

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSINDEXPART	DSSIZE	DSSIZE	
		IXDSSIZE	
	FREEPAGE	FREEPG	
		IXFREEPG	
	GBPCACHE	GBPCACHE	
		IXGBPCACH	
	IXCREATOR	OWNER	
		IXOWNER	
	IXNAME	IXNAME	
	PCTFREE	PCTFREE	
		IXPCTFREE	
	PQTY	PRIQTY	
		IXPRIQTY	
	STORNAME	IXSGNAME	
	SQTY	SECQTY	
		IXSECQTY	
	VCATNAME	VCATNAME	
SYSKEYCOLUSE	COLNAME	COLNAME	
	TBCREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	
SYSKEYS	COLNAME	COLNAME	
	IXCREATOR	OWNER	
		IXOWNER	
	IXNAME	IXNAME	
SYSPACKAGE	COLLID	SCHEMA	Trigger package
		COLLNAME	Normal package
	CREATOR	PKGOWNER	
	NAME	PKGNAME	Normal package
		TGNAME	Trigger package
	OWNER	PKGOWNER	
	PATHSCHEMAS	SCHEMA	Applied to each schema
	QUALIFIER	SCHEMA	

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSPACKDEP	BCREATOR	DBNAME	btype 'R','P'
		SCHEMA	btype 'F','O'
		TBOWNER	btype 'T','V','A','S','G'
		IXOWNER	btype ‘I'
		OWNER	btype none of the above
	BNAME	TBNAME	btype 'T','V','A','S','G'
		IXNAME	btype ‘I'
		TSNAME	btype 'R','P'
		UDFNAME	btype 'F'
		STPNAME	btype 'O'
		NAME	btype none of the above
	DCOLLID	COLLNAME	Normal package
		SCHEMA	Trigger package
	DNAME	PKGNAME	Normal package
		TGNAME	Trigger package
	DOWNER	OWNER	
SYSPARMS	NAME	STPNAME	Stored procedure
		UDFNAME	UDF
	OWNER	OWNER	
	SCHEMA	SCHEMA	
	SPECIFICNAME	STPNAME	Stored procedure
		UDFNAME	UDF
	TYPENAME	UDTNAME	If schema not SYSIBM
	TYPESCHEMA	SCHEMA	If schema not SYSIBM
SYSPLAN	CREATOR	OWNER	
	NAME	PLNNA	
	PATHSCHEMAS	SCHEMA	Applied to each schema
	QUALIFIER	OWNER	
SYSPLANDEP	BCREATOR	DBNAME	btype 'R','P'
		SCHEMA	btype 'F','O'
		TBOWNER	btype 'T','V','A','S','G'
		IXOWNER	btype ‘I'
	BNAME	TBNAME	btype 'T','V','A','S','G'
		IXNAME	btype ‘I'
		TSNAME	btype 'R','P'
		UDFNAME	btype 'F'
		STPNAME	btype 'O'
		NAME	btype none of the above
	DNAME	PLNNAME	

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSRELS	CREATOR	OWNER	
		TBOWNER	
	IXNAME	IXNAME	If non-blank
	IXOWNER	OWNER	If non-blank
		IXOWNER	If non-blank
	REFTBCREATOR	OWNER	
		TBOWNER	
	REFTBNAME	TBNAME	
	RELNAME	NAME	
	TBNAME	TBNAME	
SYSRESAUTH	GRANTEE	GRANTEE	
	GRANTOR	GRANTOR	
	NAME	COLLNAME	obtype 'C'
		NAME	obtype 'J'
		TSBPNAME	obtype 'B'
		TSNAME	obtype 'R'
		TSSGNAME	obtype 'S'
		UDTNAME	obtype 'D'
	QUALIFIER	DBNAME	obtype 'R'
		SCHEMA	obtype 'D'
SYSROUTINEAUTH	COLLID	COLLNAME	If package
	GRANTEE	GRANTEE	If authorization ID GRANTEE
		PKGNAME	If package
		PLNNAME	If plan
	GRANTOR	GRANTOR	
	SCHEMA	SCHEMA	
SYSROUTINES	COLLID	COLLNAME	
	EXTERNAL_NAME	PGMNAME	
	JAR_ID	NAME	
	JARSCHEMA	SCHEMA	
	NAME	STPNAME	stored procedure
		UDFNAME	UDF
	OWNER	OWNER	
	SCHEMA	SCHEMA	
	SOURCESCHEMA	SCHEMA	
	SPECIFICNAME	STPNAME	stored procedure
		UDFNAME	UDF

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSSCHEMAAUTH	GRANTEE	GRANTEE	
	GRANTOR	GRANTOR	
	SCHEMANAME	SCHEMA	
SYSSEQUENCEAUTH	GRANTEE	GRANTEE	
	GRANTOR	GRANTOR	
	NAME	NAME	
	SCHEMA	SCHEMA	
SYSSEQUENCES	NAME	NAME	
	OWNER	OWNER	
	SCHEMA	SCHEMA	
SYSSTOGROUP	CREATOR	CREATOR	
	KEYLABEL	SGKEYLABL	
	NAME	TSSGNAME	
	VCATNAME	VCATNAME	
SYSSYNONYMS	CREATOR	OWNER	
	NAME	TBNAME	
	TBCREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	
SYSTABAUTH	BNAME	DBNAME	
	COLLID	COLLNAME	If package
	GRANTEE	GRANTEE	If authid grantee
		PKGNAME	If package
		PLNNAME	If plan
	GRANTOR	GRANTOR	
	SCREATOR	OWNER	
	STNAME	TBNAME	
		TBOWNER	
	TCREATOR	OWNER	
		TBOWNER	
	TTNAME	TBNAME	
SYSTABCONST	CREATOR	OWNER	
	IXNAME	IXNAME	
	IXOWNER	OWNER	
		IXOWNER	
	TBCREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSTABLEPART	COMPRESS	COMPRESS	
		TSCOMPRES	
	DBNAME	DBNAME	
	DSSIZE	DSSIZE	
		TSDSSIZE	
	FREEPAGE	FREEPG	
		TSFREEPG	
	GBPCACHE	GBPCACHE	
		TSGBPCACH	
	HASHSPACE	HASHSPC	
	IXCREATOR	OWNER	
		IXOWNER	
	IXNAME	IXNAME	
	MEMBER_CLUSTER	MEMCLUS	
	PCTFREE	PCTFREE	
		TSPCTFREE	
	PCTFREE_UPD	TSPCTFUPD	
	PQTY	PRIQTY	
		TSPRIQTY	
	SQTY	SECQTY	
		TSSECQTY	
	STORNAME	TSSGNAME	
	TRACKMOD	TRACKMOD	
	TSNAME	TSNAME	
	VCATNAME	VCATNAME	

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSTABLES	APPEND	APPEND	
	AUDITING	AUDIT	
	CLUSTERTYPE	RESONDROP	
	CREATOR	OWNER	
		TBOWNER	
	DATACAPTURE	DCAPTURE	
	DBNAME	DBNAME	
	EDPROC	PGMNAME	
		EDITPROC	
	KEYLABEL	TBKEYLABL	
	NAME	TBNAME	
	SPLIT_ROWS	VOLATILE	
	TBCREATOR	OWNER	
		TBOWNER	
	TBNAME	TBNAME	
	TSNAME	TSNAME	
	VALPROC	PGMNAME	
		VALIDPROC	

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSTABLESPACE	BPOOL	TSBPNAME	
	CLOSERULE	CLOSE	
		TSCLOSE	
	COMPRESS	COMPRESS	
		TSCOMPRES	
	CREATOR	OWNER	
		TSOWNER	
	DBNAME	DBNAME	
	DSSIZE	DSSIZE	
		TSDSSIZE	
	ERASERULE	ERASE	
		TSERASE	
	FREEPAGE	FREEPG	
		TSFREEPG	
	GBPCACHE	GBPCACHE	
		TSGBPCACH	
	INSERTALG	INSALGO	
	LOCKMAX	LOCKMAX	
	LOCKRULE	LOCKSIZE	
	LOG	LOGGED	
	MAXROWS	MAXROWS	
	MEMBER_CLUSTER	MEMCLUS	
	NAME	TSNAME	
	PARTITIONS	TSPARTS	
	PCTFREE	PCTFREE	
		TSPCTFREE	
	PCTFREE_UPD	TSPCTFUPD	
	PQTY	PRIQTY	
		TSPRIQTY	
	SECQTYI	SECQTY	
		TSSECQTY	
	SEGSIZE	SEGSIZE	
	STORNAME	SGNAME	
		TSSGNAME	
	STORTYPE	SGTYPE	
		TSSGTYPE	
	TRACKMOD	TRACKMOD	
	VCATNAME	VCATNAME	

Table 20. Catalog columns and the corresponding masks (continued)

Db2 Catalog table	Catalog column	Most specific mask name	Comments
SYSTRIGGERS	NAME	TGNAME	
	OWNER	OWNER	
	SCHEMA	SCHEMA	
	TBNAME	TBNAME	
	TBOWNER	OWNER	
		TBOWNER	
	TEXT	SCHEMA	Mask trigger name
		TGNAME	Mask tab/view/synonym
		OWNER	Mask UDT/UDF/STP
		TBNAME	
		SCHEMA	
		UDTNAME	
		UDFNAME	
		STPNAME	
		COLNAME	Mask column name
SYSVIEWS	CREATOR	OWNER	
		TBOWNER	
	NAME	TBNAME	
	PATHSCHEMAS	SCHEMA	Applied to each schema
	TEXT	SCHEMA	Mask trigger name
		TGNAME	Mask tab/view/synonym
		OWNER	Mask UDT/UDF/STP
		TBNAME	
		SCHEMA	
		UDTNAME	
		UDFNAME	
		STPNAME	
		COLNAME	Mask column name
SYSVOLUMES	SGCREATOR	OWNER	
	SGNAME	TSSGNAME	
XSROBJECTS	XSROBJECTNAME	XMLSCHID	

Related concepts

"Masks" on page 273

A *mask* provides the ability to make context-sensitive changes to naming conventions, overwrite the current values of certain object attributes, and verify that object attributes conform to certain rules.

Specifying a REXX user exit for a mask overwrite value

When you define an overwrite mask, instead of explicitly specifying the overwrite value for the object attribute, you can use a REXX user exit to specify the value. Using a REXX user exit allows for additional flexibility and customization; you can define your own overwrite rules in the user exit.

Before you begin

Ensure that the data set names for the REXX user exit libraries were defined during the customization of Db2 Admin Tool. (This action is done as part of using Tools Customizer to set up Db2 Admin Tool.)

Procedure

To specify a REXX user exit for a mask overwrite value:

1. Define a REXX user exit to calculate and return a valid value for the overwrite value. Store the user exit in the appropriate REXX user exit library.

   An example REXX user exit called MYDSSIZE is included in the SAMP library ADBDSIZE. MYDSSIZE calculates and returns a value to use as the overwrite value for DSSIZE.

2. When you define the mask, use the following syntax to specify a REXX user exit as the overwrite value:

   ```
   maskname: inputmask, REXX(execname,val1,val2,...valn)
   ```

   REXX

   Indicates that a REXX user exit is to be used to calculate the overwrite value.

   execname

   The name of the REXX user exit.

   val1, val2, ... valn

   Input variables to pass as arguments to the user exit. The REXX user exit uses the arguments to perform calculations and return the value that is to be used as the overwrite value.

   Each input variable must be the name of a Db2 catalog column or a variable name with a numeric or string value, where the variable name is the name of a Db2 catalog column. If one of the input variables is not provided in the proper context, a minus sign (-) is passed to the REXX user exit as the argument.

   Restriction: If the mask is to be used for WSL cloning or the import function in Change Management, specify the input variables as Db2 catalog names that are set to numeric or string variables. If you specify a catalog name only, the variable is passed as a minus sign (-), and the REXX user exit returns a value of a minus sign (-), which indicates that masking was not applied.

   The following example mask definitions use REXX user exits as the overwrite value:

```
PRIQTY: MYDB*. MYTS*, REXX(MYPQTY, DBNAME, TSNAME, PCT= 20%)
DEFINE: MYDB*. MYTS*, REXX(MYDEFINE, DEFINE='YES')
DEFER: MYDB*. MYTS*, REXX(MYDEFER, DEFER='NO')
COMPRESS: MYDB*. MYTS*, REXX(MYCOMP,TSNAME,DBAME,COMPRESS)
SEGSIZE: MYDB*. MYTS*, REXX(MYSEG, NAME,DBNAME,SEGSIZE)
FREEPG:*.*,REXX(MYFREEPG,DBNAME,TSNAME,IXCREATOR,IXNAME)
PCTFREE:*.*,REXX(MYPCT,DBNAME='MYDBTEST',TSNAME='MYTSTEST',IXCREATOR='MYIXSCH1',IXNAME='MYIXN
AM1')
LOCKMAX:DBTEST3.TSTEST3,REXX(MYLOCKM,NAME,DBNAME)
ERASE:*.*,REXX(MYERASE,NAME,DBNAME,ERASERULE)
RESONDROP:TBCRE*.TB*,REXX(MYRODEX,DBNAME,TSNAME)
SGKEYLABL:SG1,REXX(MYKEYLB,NAME,KEYLABEL)
TBKEYLABL:TBCRE*.MYTB*,REXX(MYKEYLB,CREATOR,NAME,KEYLABEL)
```

   If the REXX user exit does not return a valid value for the overwrite value, masking is not applied, and Db2 Admin Tool processes the next specified mask.

Related concepts

“Masks” on page 273
A *mask* provides the ability to make context-sensitive changes to naming conventions, overwrite the current values of certain object attributes, and verify that object attributes conform to certain rules.

Mask recommendations

Although Db2 Admin Tool provides flexibility in specifying masks, think carefully about your masks specifications to avoid errors.

Consider the following recommendations:

- Use specific masks when possible. If you use a very broad name mask when objects in the compare scope already have that name, unexpected rows might be added to the version file when processing objects. This situation can ultimately result in an error and might terminate the compare process.

  For example, suppose that you use the following specification to mask any storage group name to ABCD:

  ```
  SGNAME: *, ABCD
  ```

  If a storage group in the scope of the comparison is already named ABCD, duplicate records will exist in the version file. These duplicate records result in an error during the compare. To avoid this situation, use one of the following more specific masks if possible:

  ```
  TSSGNAME:tablespace_schema.tablespace_name: *, ABCD
  IXSGNAME:index_schema.index_name: *, ABCD
  ```

- Consider whether using an ignore is a better option.

  For example, consider the following mask example:

  ```
  SGNAME: *, ABCD
  ```

  In this case, if you do not want to compare the storage groups at all, use an ignore. You can ignore the storage groups by using a specific ignore on STORNAME or by using a generic ignore on STORAGE.

Related concepts

“Ignores” on page 846
An *ignore* provides the ability to specify that certain fields in the Db2 catalog records are to be ignored when objects are compared. Ignores help avoid meaningless comparisons and protect those fields, called *ignore fields*, from being changed.

“Masks” on page 273
A *mask* provides the ability to make context-sensitive changes to naming conventions, overwrite the current values of certain object attributes, and verify that object attributes conform to certain rules.

Creating masks in the Change Management repository

The recommended way to define a mask is in the Change Management (CM) repository. When you define masks through CM, the masks are stored in Db2 tables in the CM repository, which makes them easy to share, manage, and recover.

The alternative to creating masks in the CM repository is to define them in a data set. (See “Creating masks in a data set” on page 309.)

Restriction: Masks that are used when you import changes through CM must be defined in the CM repository, not in a data set.

Before you begin

Before you create masks in CM, the CM repository tables must be defined as part of the process of customizing Db2 Administration Tool.

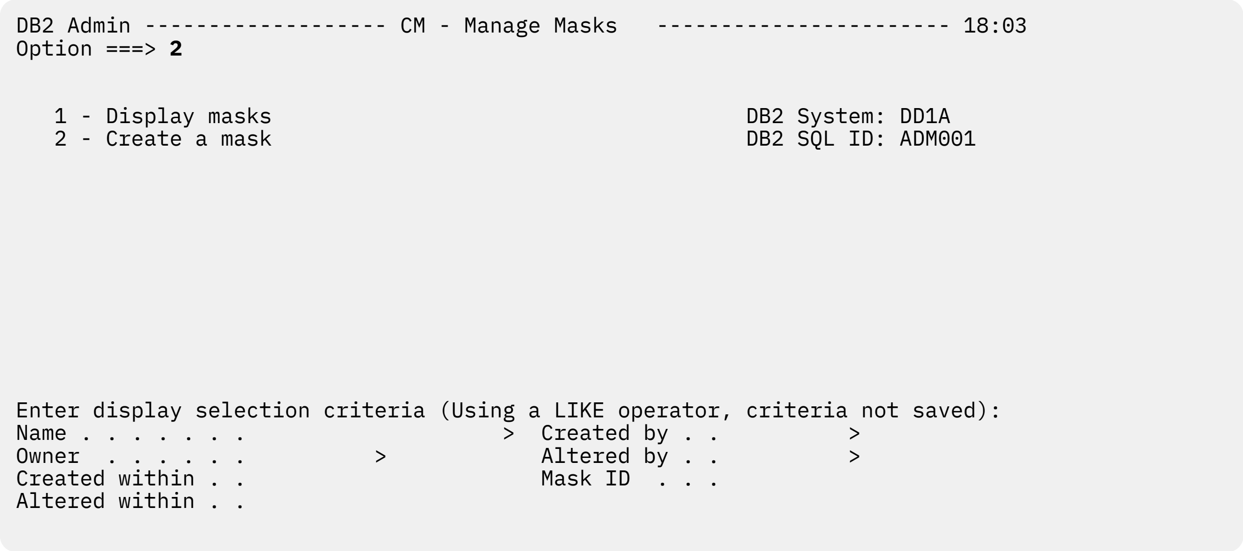
About this task

This procedure shows you how to define masks before you begin using any functions where the masks are needed. Alternatively, you can define masks while using a function, such as cloning a work statement list or reverse engineering a Db2 object. For that procedure, see “Creating masks in the CM repository while using another function” on page 308.

Procedure

To create a mask in the Change Management repository:

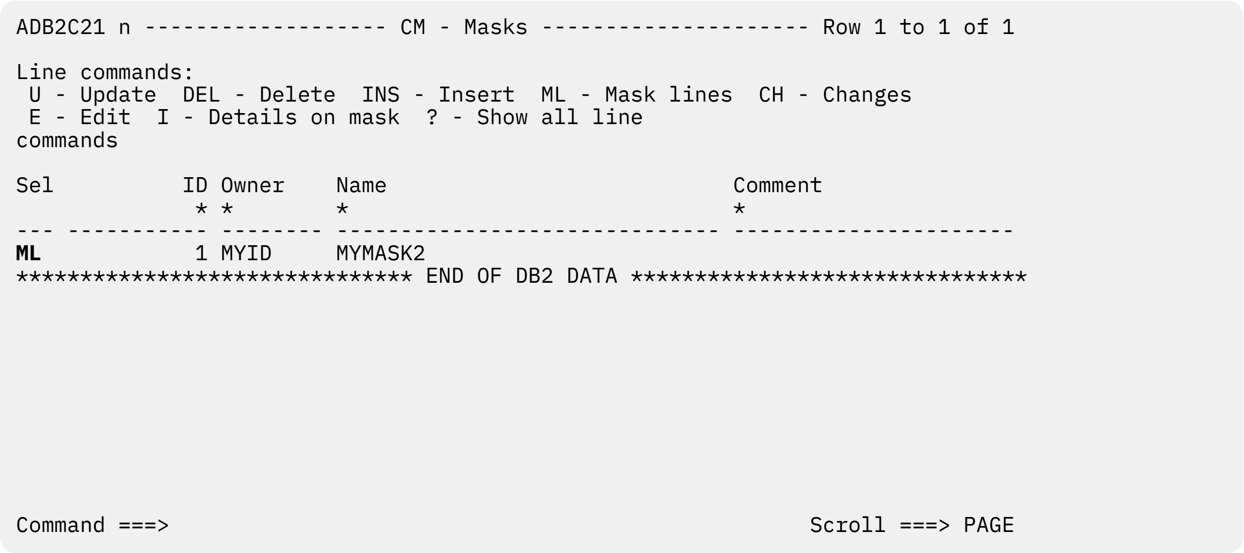
1. On the **DB2 Administration Menu (ADB2)** panel, specify option **CM**.
2. On the **Change Management (CM) (ADB2C)** panel, specify option **2**.
3. On the **Manage Masks (ADB2C2)** panel, specify option **2**:



```
DB2 Admin ------------------- CM - Manage Masks   ----------------------- 18:03
Option ===> 2


   1 - Display masks                                     DB2 System: DD1A
   2 - Create a mask                                     DB2 SQL ID: ADM001











Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .                    >  Created by . .          >
Owner  . . . . . .          >            Altered by . .          >
Created within . .                       Mask ID  . . .
Altered within . .
```

4. On the **Insert Mask (ADB2C22)** panel, specify an owner, a name for the mask row in the repository table, and optionally a comment, and press Enter.

   A message confirms that an INSERT statement was executed. That statement inserted a row with the specified name into a mask table in the CM repository.
5. Press F3 to return to the **Manage Masks (ADB2C2)** panel.
6. Specify option **1**, and press Enter.
7. On the **Masks (ADB2C21)** panel, issue the ML line command next to the row that you just created so that you can add mask lines, or mask definitions for that row:



```
ADB2C21 n ------------------- CM - Masks -------------------- Row 1 to 1 of 1

Line commands:
 U - Update  DEL - Delete  INS - Insert  ML - Mask lines  CH - Changes
 E - Edit  I - Details on mask  ? - Show all line
commands

Sel          ID Owner    Name                          Comment
              * *        *                             *
--- ----------- -------- ----------------------------- ----------------------
ML            1 MYID     MYMASK2
****************************** END OF DB2 DATA ******************************










Command ===>                                                Scroll ===> PAGE
```

8. On the **Mask Lines (ADB2C2L)** panel, define the masks. To define the first mask, complete the existing row that has a **1** in the **Sequence** column.

   For each row, specify values for the following columns:

   Req
   The mask request. Specify VER for verification masks or leave blank for name or overwrite masks.

   Type
   The mask type. Possible values are:

   - A valid name mask or overwrite mask, as listed in "Mask names" on page 275.
   - One of the following rename types: RENAMEDB, RENAMETS, RENAMETB, RENAMEIX, RENAMEGV, RENAMECOL
   - For verification masks, a valid two-character object type code, as listed in "Verification masks" on page 289.

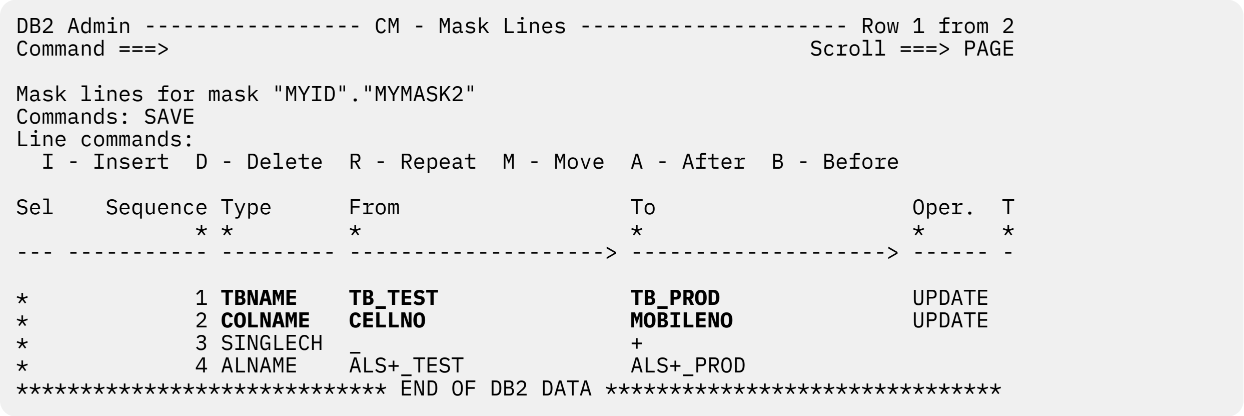
   From
   The input mask. For verification masks, specify the verification operand and verification values.

   To
   The output mask. For verification masks, specify the verification return code.

   For more information about mask names, types, and syntax, and processing see "Mask definitions" on page 274.

   To add more masks, use the **I** line command to insert more rows.

   For example, to define a mask that translates table name TB_TEST to TB_PROD and a column name from CELLNO to MOBILENO, enter the values as shown:



```
DB2 Admin ----------------- CM - Mask Lines --------------------- Row 1 from 2
Command ===>                                                Scroll ===> PAGE

Mask lines for mask "MYID"."MYMASK2"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat  M - Move  A - After  B - Before

Sel    Sequence Type      From                 To                   Oper.  T
              * *         *                    *                    *      *
--- ----------- --------- --------------------> --------------------> ------ -

*             1 TBNAME    TB_TEST              TB_PROD              UPDATE
*             2 COLNAME   CELLNO               MOBILENO             UPDATE
*             3 SINGLECH  _                    +
*             4 ALNAME    ALS+_TEST            ALS+_PROD
**************************** END OF DB2 DATA *******************************
```

You can also use the D and R line commands to delete and repeat mask lines. You can use the M line command to move the mask lines. When you specify the M line command, also specify the A or B line command to indicate which row you want the line moved after (A) or before (B). The order of the mask lines is important, because Db2 Administration Tool uses the first mask that matches. Therefore, put the most specific translation masks at the beginning and the more general ones at the end.

9. Issue the SAVE primary command to save the mask definitions.
10. Exit back to the **Manage Masks (ADB2C2)** panel.

Creating masks in the CM repository while using another function

If you are using a Db2 Admin Tool function, such as reverse engineering, migrating Db2 data, or cloning work statement lists, and need a mask that is not already defined, you can create the mask in the CM repository during the process of running that other function.

Before you begin

Change Management must be enabled.

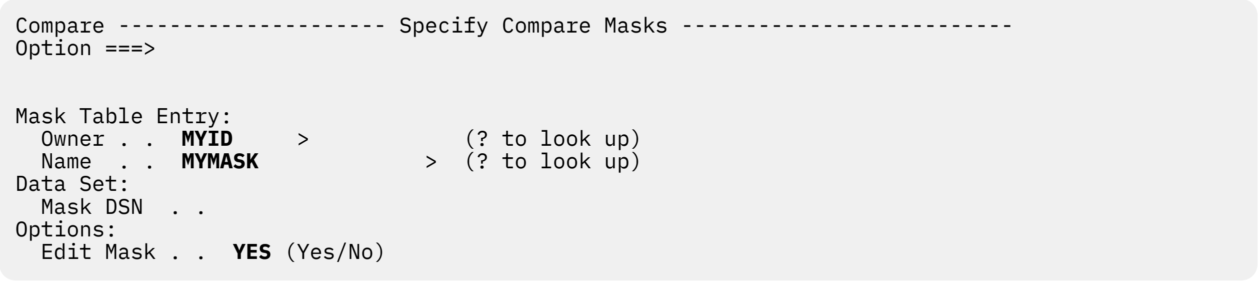
Procedure

To create a mask in the CM repository while using another function:

1. Ensure that you specify that you want to use masking (**Use Masking** = YES).

   The panel where you specify this choice varies depending on what function you are using.
2. When you are prompted to specify the masks on either the **Specify Compare Masks (GOC3)** panel or the **Specify Masks (ADB2GENM)** panel, specify the following values, and press Enter:

   - Specify an owner and name for the table row in the CM mask repository.
   - Leave the **Mask DSN** field blank.
   - In the **Edit Mask** field, specify YES.



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


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

*Figure 133. **Specify Compare Masks (GOC3)** panel*

3. On the **Insert Mask (ADB2C22)** panel, confirm the owner and name, and press Enter.

   A message confirms that an INSERT statement was executed. That statement inserted a row with the specified name into a table in the CM repository.
4. Press F3 to display the **Mask Lines (ADB2C2L)** panel.
5. Define the masks for that row in the CM repository by completing steps "8" on page 307 through "10" on page 308 (in "Creating masks in the Change Management repository" on page 305).

Creating masks in a data set

The recommended way to create masks is to define them in the Change Management repository. However, if you do not have Change Management enabled, you can still use masks by defining them in a data set.

About this task

The easiest way to define masks in a data set is to do so in the context of using one of the following functions for which you want the mask:

- Generate SQL to reverse engineer Db2 objects
- Clone a work statement list (WSL)
- Migrate objects, data, or catalog statistics

Procedure

To create masks in a data set:

1. On one of the following panels, in the **Use Masking** field, specify `Yes` and press Enter:

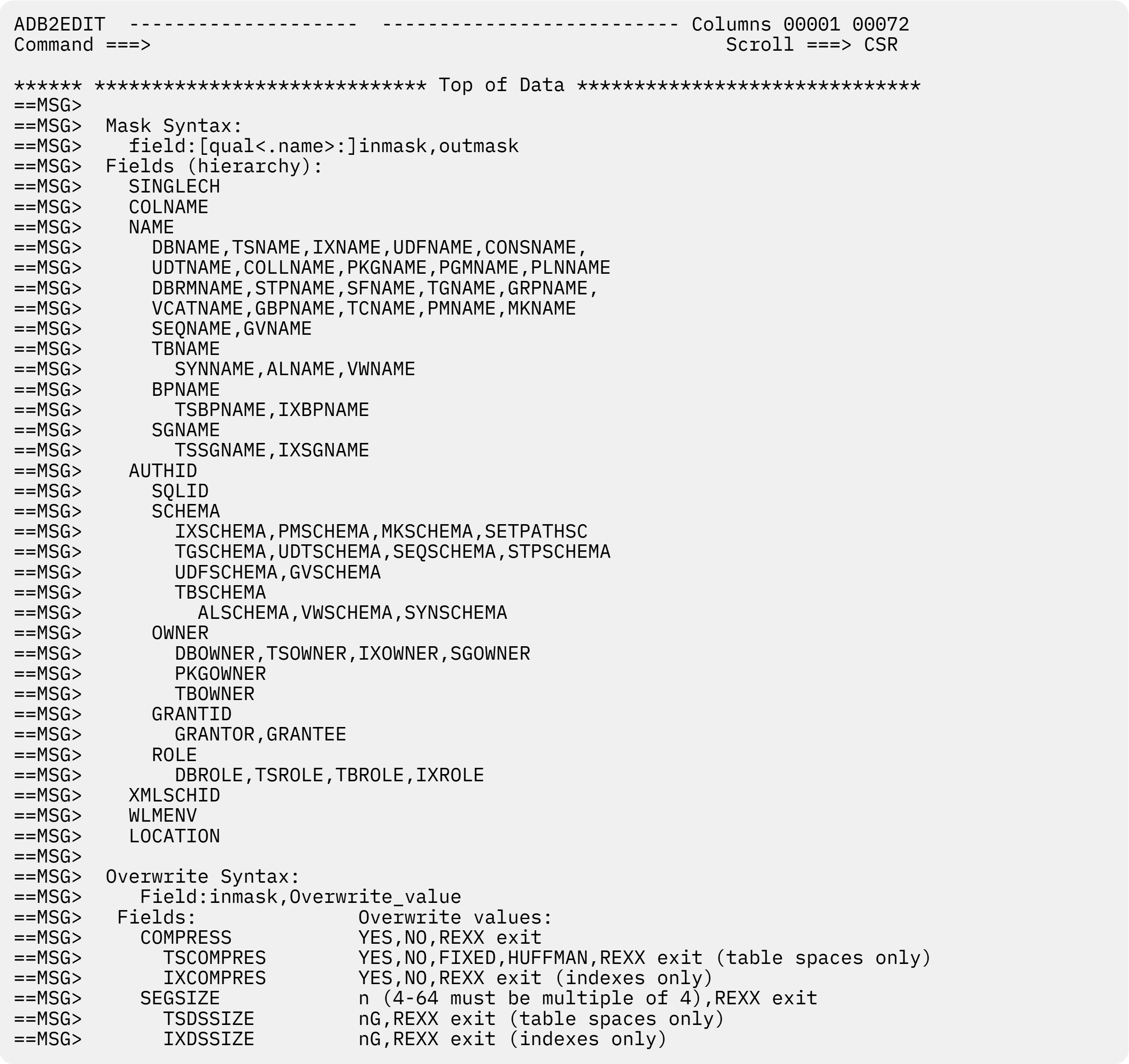
   - **Generate SQL from DB2 catalog (ADB2GENB)** panel
   - **Clone Work Statement List (ADB2W1Q)** panel
   - **Migrate Parameters (ADB28M)** panel

   The panel that you use depends on which function you are using.
2. On the **Specify Masks (ADB2GENM)** panel, specify a data set name. This data set is to store the mask definitions.

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

   - A fixed-block sequential data set `(RECFM=Fx)`
   - A member of a partitioned data set with a record length of `80 (LRECL=80)`
3. In the **Edit Mask** field, specify YES, and press Enter.

   The data set is created, and an ISPF edit panel is displayed where you can edit the data set. This mask data set is pre-populated with a list of mask names and their hierarchy in the MSG lines, as shown in the following figures:

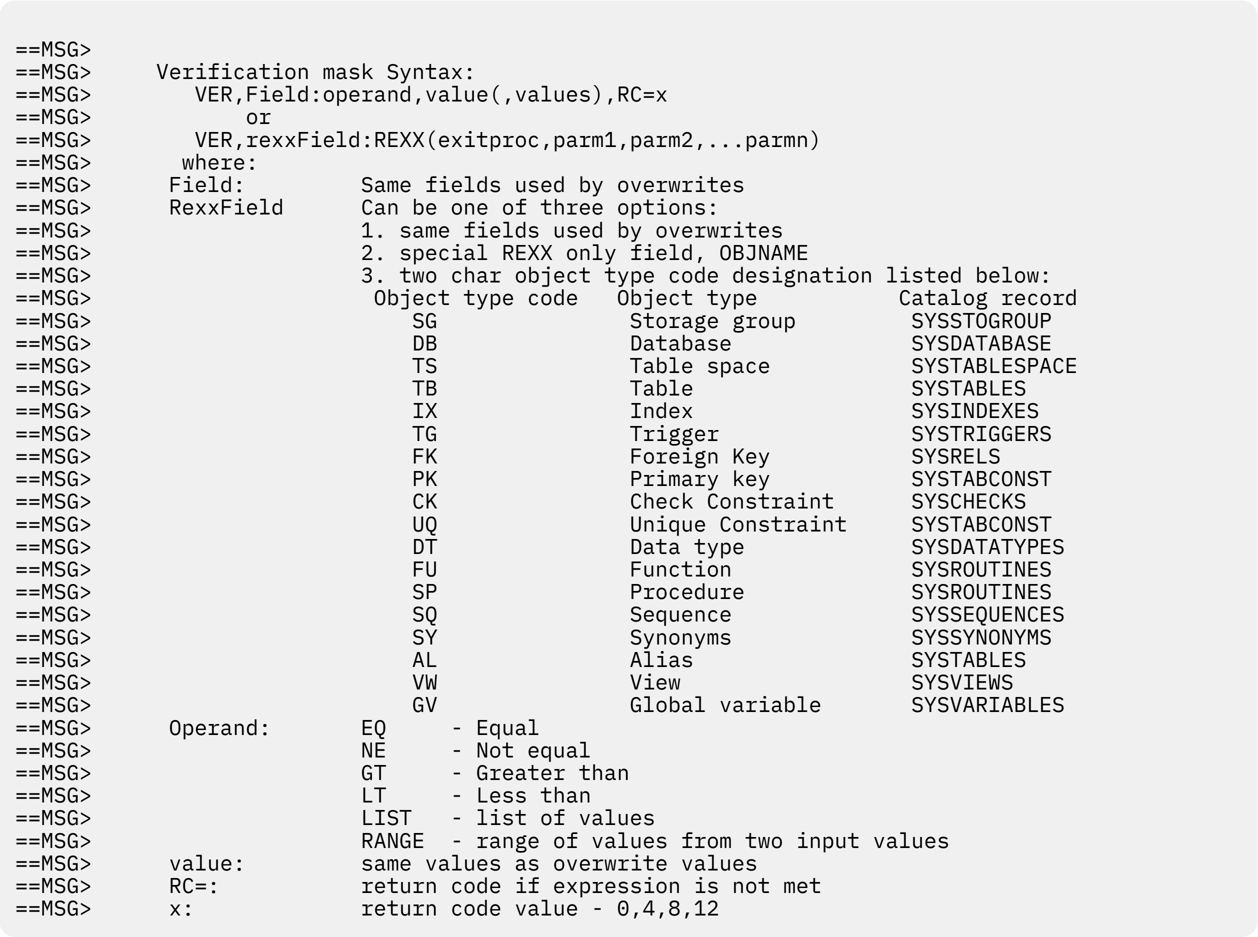


```
ADB2EDIT  --------------------  -------------------------- Columns 00001 00072
Command ===>                                                  Scroll ===> CSR

****** ***************************** 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>
==MSG>  Overwrite Syntax:
==MSG>     Field:inmask,Overwrite_value
==MSG>   Fields:              Overwrite values:
==MSG>     COMPRESS           YES,NO,REXX exit
==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)
```

Figure 134. ***Edit Masks (ADB2EDIT)*** *panel, part 1*

```
==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,REXX exit (table spaces only)
==MSG>     SGKEYLABL          string,NO,NOKEYLABEL,REXX exit (stogroup only)
==MSG>     TBKEYLABL          string,NO,NOKEYLABEL,REXX exit (tables only)
```

*Figure 135. **Edit Masks (ADB2EDIT)** panel, Part 2*

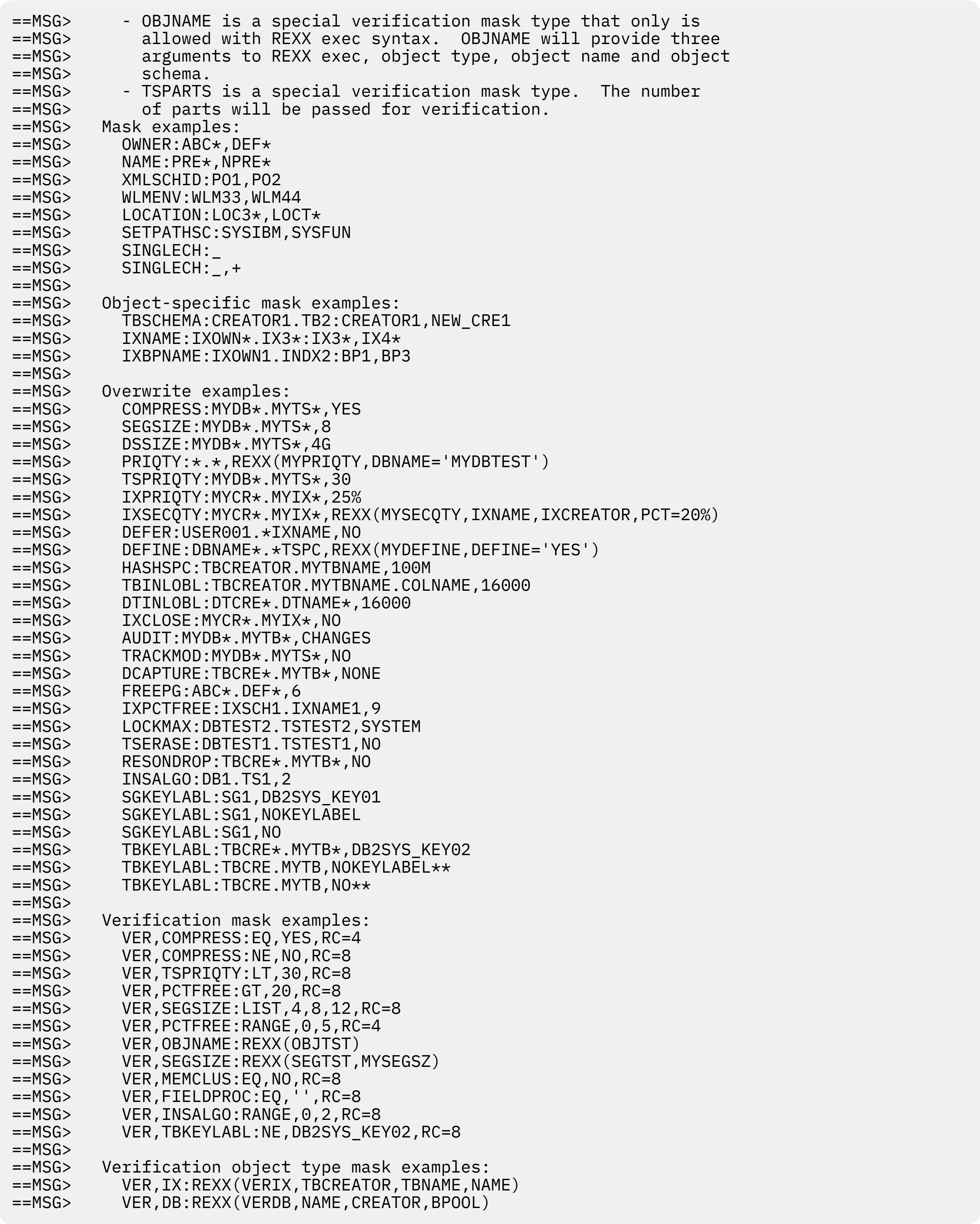


```
==MSG>
==MSG>     Verification mask Syntax:
==MSG>        VER,Field:operand,value(,values),RC=x
==MSG>            or
==MSG>        VER,rexxField:REXX(exitproc,parm1,parm2,...parmn)
==MSG>       where:
==MSG>      Field:         Same fields used by overwrites
==MSG>      RexxField      Can be one of three options:
==MSG>                     1. same fields used by overwrites
==MSG>                     2. special REXX only field, OBJNAME
==MSG>                     3. two char object type code designation listed below:
==MSG>                      Object type 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>                         CK               Check Constraint      SYSCHECKS
==MSG>                         UQ               Unique Constraint     SYSTABCONST
==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>      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
```

Figure 136. ***Edit Masks (ADB2EDIT)*** *panel, Part 3*

```
==MSG>   Notes:
==MSG>     - n is a integer value
==MSG>     - n% is the integer percentage of the current attribute value
==MSG>     - REXX exit takes format of REXX(myexit,val1,val2...valn) where
==MSG>       valn is the name of DB2 catalog field (such as PARTITIONS) or
==MSG>       a variable with numeric/string value (such as BPOOL= 'BP1').
==MSG>       + in col 72 indicates continuation of Rexx exit on next line
==MSG>     - To support/migrate DB2V8 masking input,OWNER,TBOWNER and
==MSG>       IXOWNER will mask both owner and schema fields.SCHEMA,
==MSG>       TBSCHEMA and IXSCHEMA will be applied to schema fields only.
==MSG>     - 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>[,<escape character>]
==MSG>       where the single character in a mask specification represents
==MSG>       any character at that position. If the specified escape
==MSG>       character precedes the specified single character, then the
==MSG>       single character is treated as literal.
==MSG>     - The view, alias and synonym masks (both name and
==MSG>       schema/owner) will only apply to the CREATE statement for
==MSG>       these objects (e.g. VWNAME only valid for CREATE VIEW).
==MSG>       All other usages of these names and schemas are vague and
==MSG>       can refer also to table names and schemas.  These other
==MSG>       usages can only be masked by TBNAME for name and TBSCHEMA
==MSG>       for schema; therefore, it is recommended to use both VWNAME
==MSG>       and TBNAME if view names are being changed for both CREATE
==MSG>       VIEW statement and SQL that uses this view.
==MSG>     - Use caution when specifying mask field SEGSIZE. This mask
==MSG>       field might cause changes to the table space type. For
==MSG>       example, specifying the SEGSIZE mask might convert a
==MSG>       partitioned table space to a range-partitioned universal
==MSG>       table space (UTS). If a table in a UTS has a partitioned
==MSG>       index and the partitioned index needs to be recreated, DB2
==MSG>       might generate SQLCODE=-662 during execution.
==MSG>     - The following masks can not have the object-specific
==MSG>       qualifiers listed in the mask syntax:
==MSG>        NAME, SCHEMA, SETPATHSC, DBNAME, COLLNAME, SFNAME, GRANTID,
==MSG>        GRANTOR, GRANTEE, ROLE, DBROLE, TSROLE, TBROLE, IXROLE,
==MSG>        GBPNAME, TCNAME, XMLSCHID, AUTHID, SQLID, SGNAME, OWNER,
==MSG>         OWNER, BPNAME, PLNNAME and SINGLECH.
==MSG>     - 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>     - Verification masks are only valid with the GEN and compare
==MSG>       process.  A warning stating that verification masks will
==MSG>       be ignored will be issued for all other processes that
==MSG>       allow masking.
```

*Figure 137. **Edit Masks (ADB2EDIT)** panel, Part 4*



```
==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>     - TSPARTS is a special verification mask type.  The number
==MSG>       of parts will be passed for verification.
==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>     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,COMPRESS:NE,NO,RC=8
==MSG>     VER,TSPRIQTY:LT,30,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:EQ,'',RC=8
==MSG>     VER,INSALGO:RANGE,0,2,RC=8
==MSG>     VER,TBKEYLABL:NE,DB2SYS_KEY02,RC=8
==MSG>
==MSG>   Verification object type mask examples:
==MSG>     VER,IX:REXX(VERIX,TBCREATOR,TBNAME,NAME)
==MSG>     VER,DB:REXX(VERDB,NAME,CREATOR,BPOOL)
```

*Figure 138. **Edit Masks (ADB2EDIT)** panel, Part 5*

4. Issue the SAVE command to save the changes to the data set.
5. Exit out of the ISPF editor.

Specifying masks

You can use masks when you generate SQL to reverse engineer Db2 objects, clone a work statement list (WSL), or migrate objects, data, or catalog statistics.

About this task

Specifying masks is a substep of one of the following procedures:

- "Generating SQL to re-create a Db2 object (reverse engineering)" on page 357
- "Cloning a WSL" on page 543

Procedure

To specify masks:

1. Ensure that one of the following panels is opened:

   - **Generate SQL from DB2 catalog (ADB2GENB)** panel
   - **Clone Work Statement List (ADB2W1Q)** panel

   The panel that you use depends on which function you are using.
2. In the **Use Masking** field, specify `Yes` and press Enter:
3. On the **Specify Masks (ADB2GENM)** panel, specify the mask to use by completing one of the following set of steps:

   To specify a mask that is defined in a data set:

   a) Specify the name of the data set that contains the masks that you want to use. If the specified data set does not exist, 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=Fx)`
      - 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.

   c) Press Enter.

      If you specified that you wanted to edit the mask, an ISPF edit panel opens. On this panel, you can make changes to the mask definitions. When you are done making changes, issue the SAVE command and exit out of the ISPF editor.

   To specify a mask that is defined in a table in the Change Management database:

   a) Specify the owner and the name of the mask table entry in the **Owner** and **Name** fields.

      (The **Mask Table Entry** fields where you can specify an owner and name are displayed only if Change Management is enabled on your system.)

   b) Leave the **Mask DSN** field blank.

      (If you specify both a data set and a mask table entry, a data set is used.)

   c) In the **Edit Mask** field, specify whether you want to edit the mask definitions.

   d) Press Enter.

      If you specify a mask table entry that does not exist, that table entry is created for you in the Change Management database.

      If you specified that you wanted to edit the mask, the **Mask Lines (ADB2C2L)** panel opens. On this panel, you can make changes to the mask definitions. For more information about how change the definitions, see "Creating masks in the Change Management repository" on page 305

```
DB2 Admin ----------------------- Specify Mask ----------------------------
Option ===>


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

What to do next

Finish the remaining steps of one of the following the procedures for which you wanted a mask:

- “Generating SQL to re-create a Db2 object (reverse engineering)” on page 357
- “Cloning a WSL” on page 543

Displaying masks in the Change Management database

You can view a list of the mask table entries that are already defined in the Change Management database.

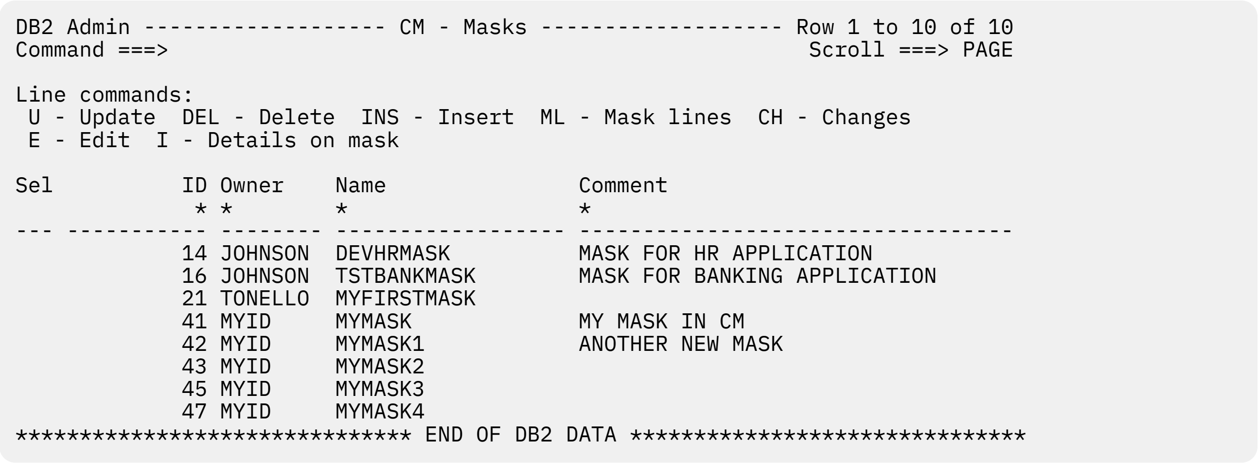
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option **CM**, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option **2**, and press Enter.
3. Optional: At the bottom of the **Manage Masks (ADB2C2)** panel, specify search criteria to filter the masks that you want to view:

```
DB2 Admin ------------------- CM - Manage Masks   ----------------------- 18:03
Option ===>


   1 - Display masks                                     DB2 System: DD1A
   2 - Create a mask                                     DB2 SQL ID: ADM001










Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .                    >  Created by . .          >
Owner  . . . . . .          >            Altered by . .          >
Created within . .                       Mask ID  . . .
Altered within . .
```

4. Specify option **1**, and press Enter.

   The list of mask sets that are defined in Change Management are displayed on the **Masks (ADB2C21)** panel.



```
DB2 Admin ------------------- CM - Masks ------------------- Row 1 to 10 of 10
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  DEL - Delete  INS - Insert  ML - Mask lines  CH - Changes
 E - Edit  I - Details on mask

Sel          ID Owner    Name               Comment
              * *        *                  *
--- ----------- -------- ------------------ ----------------------------------
             14 JOHNSON  DEVHRMASK          MASK FOR HR APPLICATION
             16 JOHNSON  TSTBANKMASK        MASK FOR BANKING APPLICATION
             21 TONELLO  MYFIRSTMASK
             41 MYID     MYMASK             MY MASK IN CM
             42 MYID     MYMASK1            ANOTHER NEW MASK
             43 MYID     MYMASK2
             45 MYID     MYMASK3
             47 MYID     MYMASK4
******************************* END OF DB2 DATA *******************************
```

Masks that are defined in data sets outside of Change Management are not displayed, because they are not stored in the Change Management database. However, those masks are still available to use in other Db2 Admin Tool functions.

To view the mask definitions for a particular mask table entry, use the ML line command. You can use the other line commands to view information about the mask table entries, edit them, and define new masks.

Related tasks

“Editing and deleting masks in the Change Management database” on page 317
Mask definitions in Change Management are stored in rows in a table. Each row, or entry, in a mask table has a name and contains a set of mask lines or definitions. You can edit or delete those mask definitions.

Editing and deleting masks in the Change Management database

Mask definitions in Change Management are stored in rows in a table. Each row, or entry, in a mask table has a name and contains a set of mask lines or definitions. You can edit or delete those mask definitions.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option **CM**.
2. On the **Change Management (CM) (ADB2C)** panel, specify option **2**.
3. On the **Manage Masks (ADB2C2)** panel, specify option **1**.
4. If you want to delete a mask table entry, on the **Masks (ADB2C21)** panel, issue the DEL line command next to the mask entry that you want to delete.
5. If you want to edit a mask table entry, on the **Masks (ADB2C21)** panel, issue one of the following line commands next to the mask entry that you want to edit:

   ML

   Opens the **Mask Lines (ADB2C2L)** panel where you can add, delete and change mask definitions that are associated with the table entry. For information about how to add and change these mask definitions, see step “8” on page 307 (in “Creating masks in the Change Management repository” on page 305).

   E

   Opens an ISPF edit panel where you can make changes to the mask definitions that are associated with the table entry. The edit panel includes a list of possible mask names and their hierarchy.

   After you make your changes on **Mask Lines (ADB2C2L)** panel or the ISPF edit panel, press PF3 to save your changes and return to the **Masks (ADB2C21)** panel.

SQL statements

Db2 Admin Tool can issue, build, and run SQL statements.

This information describes how to use Db2 Admin Tool to perform the following tasks:

- Issue dynamic SQL statements from your screen, from a data set, or from program file
- Build and run SQL SELECT, INSERT, UPDATE, and DELETE statements interactively by using line commands
- Run the following SQL statements by entering required parameters: CREATE, DROP, LABEL ON, COMMENT ON, GRANT, and REVOKE

  The two panels for this function are also used from the system catalog panels, where they are shown when a line command is issued against an object. When invoked in this way, the object names contain the object name from the catalog.

Running SQL statements from screen input

You can specify free-form SQL statements on your screen and run them dynamically.

Procedure

To run SQL statements from screen input:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.
2. On the **Execute SQL Statements (ADB22)** panel, specify option 1, and press Enter:

```
ADB22 min -------------------- Execute SQL Statements ------------------- 01:36
Option ===> 1

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

3. On the **Edit/Run SQL Statement (ADBPMESQ)** panel, enter the SQL statement that you want to run between columns 1 and 72 by using ISPF edit commands. Do not use line numbers. If you want a line to be ignored, start that line with the SQL comment characters (--).

   By default, any SQL statement that you enter is converted to uppercase. To disable this behavior, specify the CAPS OFF primary command.

```
ADBPMESQ  ------------ Edit/Run SQL Statement ------------ Columns 00001 00072
Command ===>                                                  Scroll ===> CSR

****** ***************************** Top of Data ******************************
==MSG> Use command EXEC to run the SQL statement and return to the edit session
==MSG> or use command CANCEL to exit the edit session without running the SQL
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** **************************** Bottom of Data ****************************
```

4. Run the statement by taking one of the following actions:

   - Enter the EXEC primary command. Use this command if you want to return to the edit panel after you run the statement.
   - Press PF3. Use PF3 if you want to end the session after you run the statement. The statement is saved in a temporary data set. If you did not make any changes to the statement, you are prompted whether you want to run the statement.

If you do not want to save or run the statement, use the CANCEL command.

If an SQL SELECT statement returns rows, the result is shown on the default table display panel.

After the SQL statement is run, a message is displayed on the top left side of the panel to say that the statement was executed. For example:

```
ADB22 min -------------------- Execute SQL Statements ------------------- 16:30
Option ===>
INSERT stmt executed
   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

While this message is displayed, press PF1 to see any additional information. For UPDATE, INSERT, and DELETE statements, the number of affected rows is displayed. For example:

```
ADB260I 1 row(s) affected by the INSERT statement
```

Running SQL statements from a data set

You can run SQL statements that are stored in a data set.

Procedure

To run SQL statements from a data set:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.
2. On the **Execute SQL Statements (ADB22)** panel, specify option 2, and press Enter:

```
 ADB22 min -------------------- Execute SQL Statements ------------------- 01:36
 Option ===> 2

    1 - Edit/run SQL statements                           DB2 System: DD1A
    2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
    3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
    4 - Create/drop/label/comment on objects
    5 - Grant/revoke privileges on objects
```

3. On the **Run or Explain SQL Statements (ADB222)** panel, specify the data set name that contains the SQL statements that you want to run:

```
DB2 Admin ---------------- Run or Explain SQL Statements ---------------- 17:44
Option ===>


   1 - Run SQL statements from a data set                DB2 System: DD1A
        EDIT first ===> YES (Yes/No)                     DB2 SQL ID: ADM001
   2 - Run or Explain SQL located in a program
        Program type ===>   (1=COBOL, 2=PL/I)

ISPF library:
   Project ===>
   Group   ===>           ===>           ===>           ===>
   Type    ===>
   Member  ===>                    (blank for member selection list)

Other partitioned or sequential data set:
   Data Set Name  ===>
   Volume Serial  ===>             (if not cataloged)

Alternative pre-allocated DD name:
   DD name ===>                    (use ddname(member) for members)
```

You can specify one of the following types as the input data set:

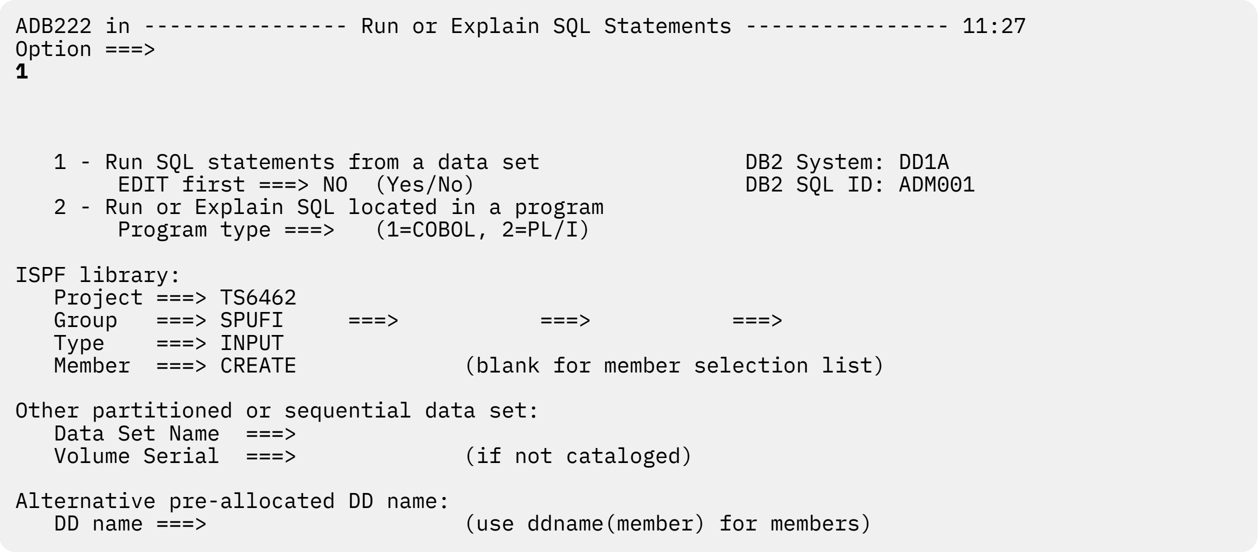
- An ISPF library

- A partitioned or sequential data set
- A pre-allocated DD name

Restrictions: The following restrictions apply to the input data set:

- If the record format (RECFM) is either F or FB and the logical record length (LRECL) is either 79 or 80, Db2 Admin Tool assumes that the last 8 bytes of each record are for sequence numbers. Therefore, do not use the last 8 columns of each record to store SQL statements. Otherwise, if the logical record length (LRECL) is not 79 nor 80, Db2 Admin Tool assumes that all of the columns of each record are for SQL statements.
- If the record format (RECFM) is either V or VB, Db2 Admin Tool checks whether the content in columns 1 through 8 of the first record is numeric. If the content is numeric, Db2 Admin Tool assumes that the first 8 bytes of each record are for sequence numbers. Otherwise, if the content of columns 1 through 8 is not numeric, Db2 Admin Tool assumes that all columns are for SQL statements.

4. In the **EDIT first** field, specify whether you want to edit the data set before running the SQL statements.
5. Specify option 1 (**Run SQL statements from a data set**), and press Enter:

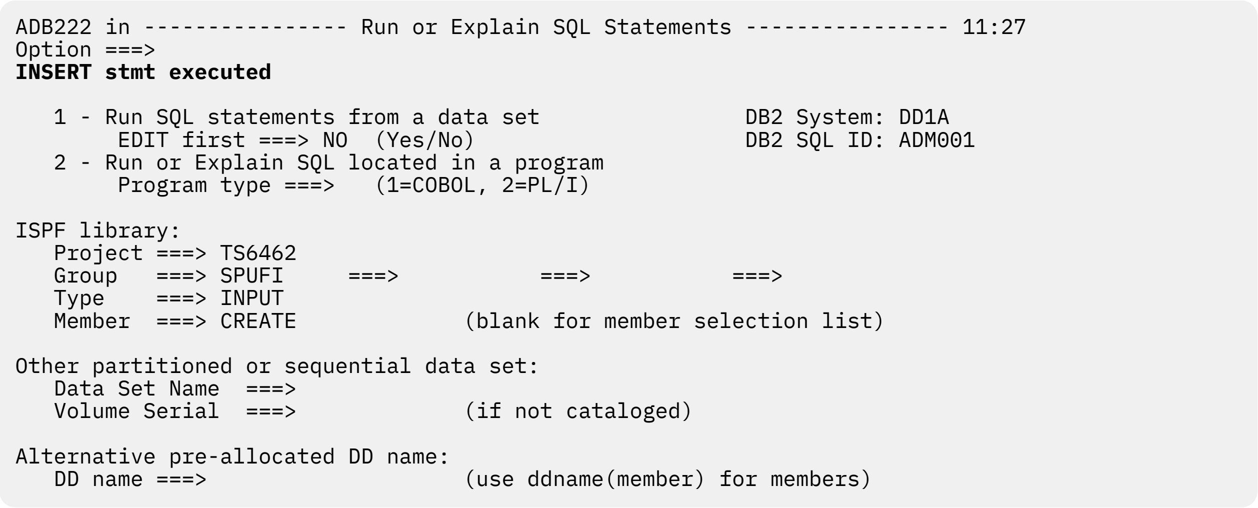


```
ADB222 in ---------------- Run or Explain SQL Statements ---------------- 11:27
Option ===>
1



   1 - Run SQL statements from a data set                DB2 System: DD1A
        EDIT first ===> NO  (Yes/No)                     DB2 SQL ID: ADM001
   2 - Run or Explain SQL located in a program
        Program type ===>   (1=COBOL, 2=PL/I)

ISPF library:
   Project ===> TS6462
   Group   ===> SPUFI     ===>           ===>           ===>
   Type    ===> INPUT
   Member  ===> CREATE             (blank for member selection list)

Other partitioned or sequential data set:
   Data Set Name  ===>
   Volume Serial  ===>             (if not cataloged)

Alternative pre-allocated DD name:
   DD name ===>                    (use ddname(member) for members)
```

If you specified Yes in the **EDIT first** field, the data set is opened in ISPF edit mode where you can then edit the statements. Press End in the edit session to run the SQL statements.

Otherwise, if you specified No in the **EDIT first** field, the statements are executed immediately.

After the SQL statement is run, a message is displayed on the top left side of the panel to say that the statement was executed. For example:



```
ADB222 in ---------------- Run or Explain SQL Statements ---------------- 11:27
Option ===>
INSERT stmt executed

   1 - Run SQL statements from a data set                DB2 System: DD1A
        EDIT first ===> NO  (Yes/No)                     DB2 SQL ID: ADM001
   2 - Run or Explain SQL located in a program
        Program type ===>   (1=COBOL, 2=PL/I)

ISPF library:
   Project ===> TS6462
   Group   ===> SPUFI     ===>           ===>           ===>
   Type    ===> INPUT
   Member  ===> CREATE             (blank for member selection list)

Other partitioned or sequential data set:
   Data Set Name  ===>
   Volume Serial  ===>             (if not cataloged)

Alternative pre-allocated DD name:
   DD name ===>                    (use ddname(member) for members)
```

While this message is displayed, press PF1 to see any additional information. For UPDATE, INSERT, and DELETE statements, the number of affected rows is displayed. For example:

```
ADB260I 1 row(s) affected by the INSERT statement
```

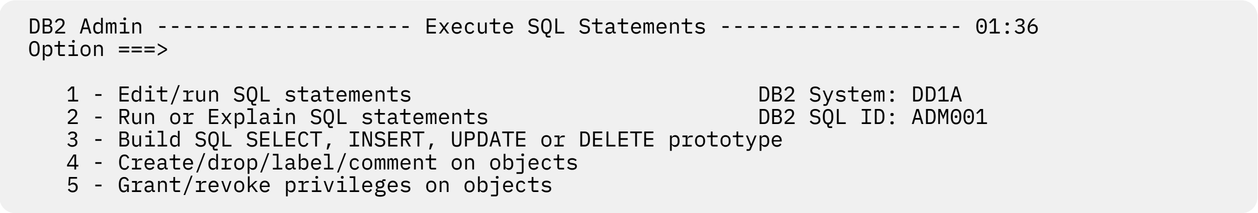
Running or explaining SQL statements from a program file

You can run or explain SQL statements that are in a program file.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.

   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure,

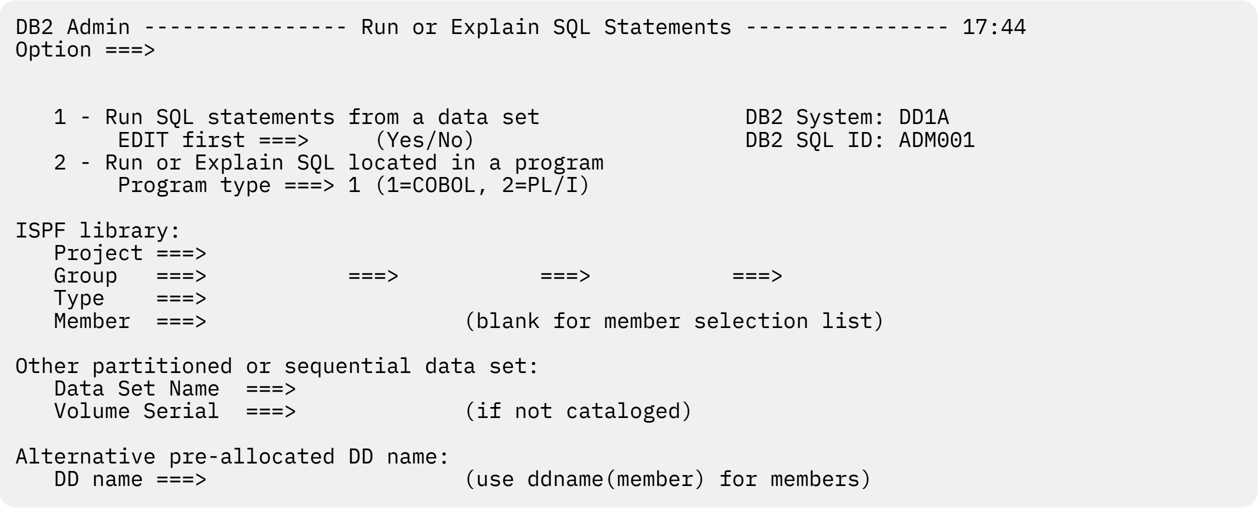


```
 DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
 Option ===>

    1 - Edit/run SQL statements                          DB2 System: DD1A
    2 - Run or Explain SQL statements                    DB2 SQL ID: ADM001
    3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
    4 - Create/drop/label/comment on objects
    5 - Grant/revoke privileges on objects
```

*Figure 139. **Execute SQL Statements (ADB22)** panel*

2. Specify option 2, and press Enter.

   The **Execute SQL Statements from a Data Set (ADB222)** panel is displayed, as shown in the following figure:



```
DB2 Admin ---------------- Run or Explain SQL Statements ---------------- 17:44
Option ===>


   1 - Run SQL statements from a data set                DB2 System: DD1A
        EDIT first ===>     (Yes/No)                     DB2 SQL ID: ADM001
   2 - Run or Explain SQL located in a program
        Program type ===> 1 (1=COBOL, 2=PL/I)

ISPF library:
   Project ===>
   Group   ===>           ===>           ===>           ===>
   Type    ===>
   Member  ===>                    (blank for member selection list)

Other partitioned or sequential data set:
   Data Set Name  ===>
   Volume Serial  ===>             (if not cataloged)

Alternative pre-allocated DD name:
   DD name ===>                    (use ddname(member) for members)
```

Figure 140. ***Execute SQL Statements from a Data Set (ADB222)*** *panel*

3. Specify option 2 to specify that the SQL statements to run or explain are in a program file and specify the type of program, and press Enter.

   The types of programs are valid:

   1
   COBOL

   2
   PL/I

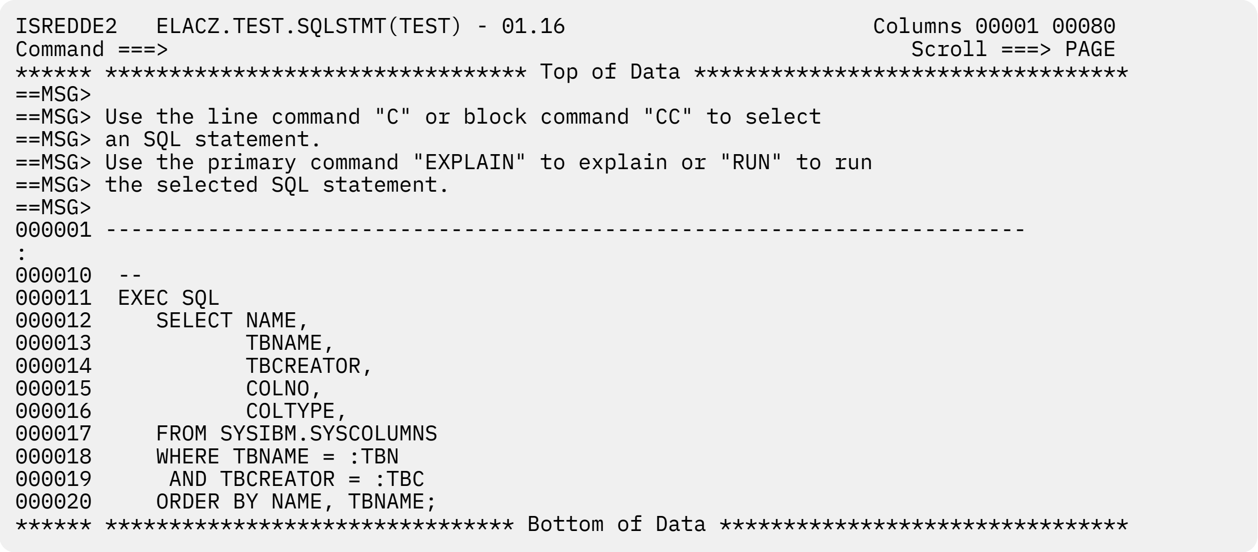
   If you specify the program type as a parameter when you issue the RUN or EXPLAIN primary command for the SQL statement, the parameter for the type overrides the value that is set in the **Program type** field.

4. Specify the data set name that contains the program, and press Enter.

   The input data set can be specified as one of the following types:

   - An ISPF library
   - A partitioned or sequential data set
   - A pre-allocated ddname

   The program file is displayed, as shown in the example in the following figure:



```
ISREDDE2   ELACZ.TEST.SQLSTMT(TEST) - 01.16                        Columns 00001 00080
Command ===>                                                          Scroll ===> PAGE
****** ********************************* Top of Data **********************************
==MSG>
==MSG> Use the line command "C" or block command "CC" to select
==MSG> an SQL statement.
==MSG> Use the primary command "EXPLAIN" to explain or "RUN" to run
==MSG> the selected SQL statement.
==MSG>
000001 ------------------------------------------------------------------------
:
000010  --
000011  EXEC SQL
000012     SELECT NAME,
000013            TBNAME,
000014            TBCREATOR,
000015            COLNO,
000016            COLTYPE,
000017     FROM SYSIBM.SYSCOLUMNS
000018     WHERE TBNAME = :TBN
000019      AND TBCREATOR = :TBC
000020     ORDER BY NAME, TBNAME;
****** ******************************** Bottom of Data ********************************
```

Figure 141. Example of selecting SQL statements in a program to run or explain

5. Use the C line command or the CC block command to select the SQL statement to run or explain.

   Only one SQL statement can be selected at a time.

   Restriction: The following SQL statements cannot be run or explained:

   - ALLOCATE CURSOR
   - ASSOCIATE LOCATOR
   - BEGIN DECLARE SECTION and END DECLARE SECTION
   - CALL
   - CLOSE
   - CONNECT
   - DECLARE STATEMENT, DECLARE TABLE, DECLARE VARIABLE
   - all DESCRIBE statements
   - EXECUTE and EXECUTE IMMEDIATE
   - FETCH
   - FREE LOCATOR and HOLD LOCATOR
   - INCLUDE
   - OPEN
   - PREPARE
   - SIGNAL SQLSTATE
   - VALUES
   - WHENEVER
   - --#SET ROWS_FETCH, --#SET ROWS_OUT, --#SET TERMINATOR
6. Issue the RUN primary command to run the statement or the EXPLAIN primary command to explain the statement.
7. Specify the values for every host variable in the SQL statement in the pop-window that is displayed. Enter the values for character host variables in single quotation marks. If you leave the value of host variable blank, the host variable is removed from the statement.
8. Exit the edit session to have the primary command executed.

   Tip: If you have changed the selected statement but do not want to save the changes in the program file, choose CANCEL when you are prompted to exit the edit session. The updated statement is executed, but the program file is not changed.

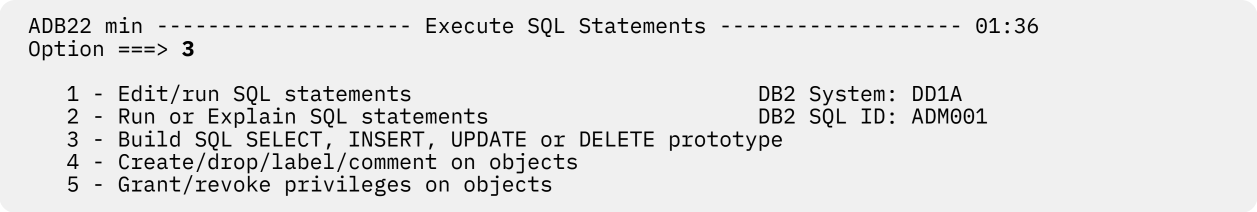
Building SQL SELECT, INSERT, UPDATE, and DELETE prototypes

The prototyping facility in Db2 Admin Tool helps you to build dynamic SQL statements without requiring you know the exact syntax. Using prototyping, you can build SQL SELECT, INSERT, UPDATE, and DELETE statements interactively.

Procedure

To build SQL SELECT, INSERT, UPDATE, and DELETE prototypes:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.
2. On the **Execute SQL Statements (ADB22)** panel, specify option 3, and press Enter:



```
ADB22 min -------------------- Execute SQL Statements ------------------- 01:36
Option ===> 3

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

3. On the **Build SQL Prototype: Search Objects (ADB223)** panel, optionally specify the schema and name of the object (table, view, or alias) for which you want to build a statement, and press Enter. If you want to choose from a list of all tables, views, and aliases, do not specify a schema or name.

   ```
   ADB223 in ------------- Build SQL Prototype: Search Objects ------------- 06:22

    Enter/verify:
     Schema  . . . ________ >            (optional, default is SMITHJR)
     Name  . . . . __________________ >  (optional)
   ```

   The list of objects that match the search criteria are displayed on the **Tables, Views, and Aliases (ADB223T)** panel.
4. Specify one of the following line commands next to the object for which you want to build a statement:

   SEL
   SELECT statement

   DEL
   DELETE statement

   INS
   INSERT statement

   UPD
   UPDATE statement

   A **Build SQL Prototype** panel is displayed with a partially built SQL statement at the top.
5. Build your SQL statement by using the available line commands and press Enter to run the statement.

   For help with the commands, see “Build SQL Prototype panel” on page 327 or the online help.

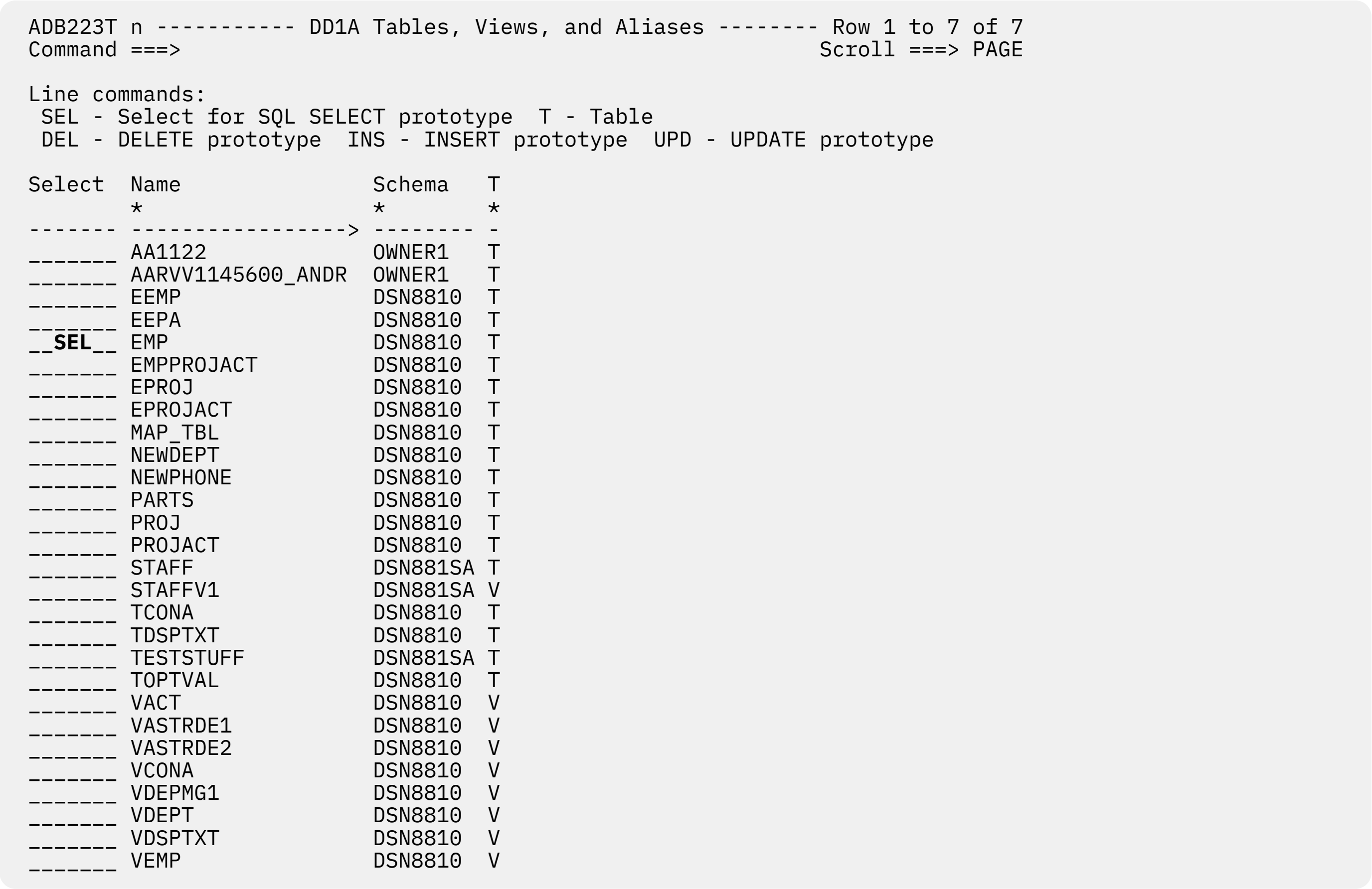
   You can use the EDIT command to capture the statement and store it in a data set.

Examples

Example of building a SELECT statement

For this example, assume that you want to build a SELECT statement that returns the name and department number of all employees with a salary greater than $30,000.

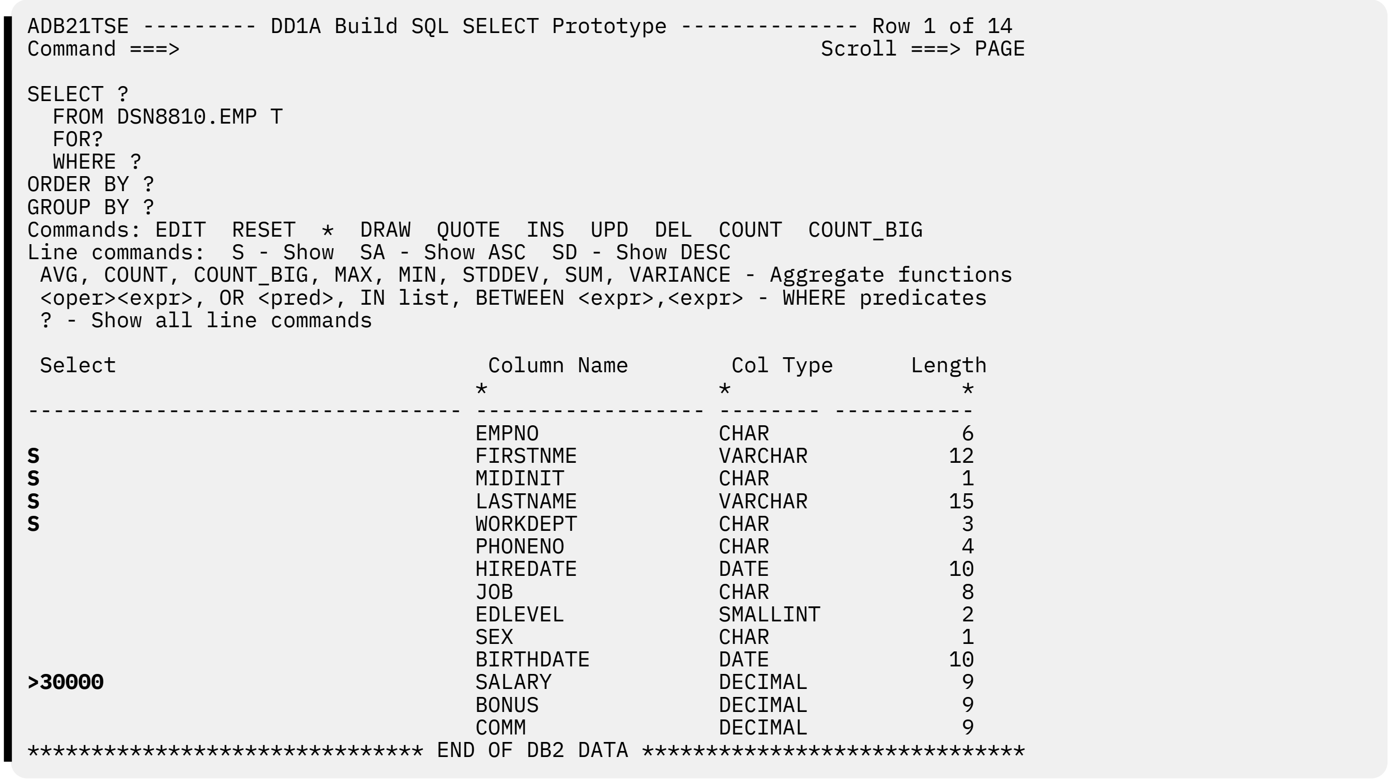
1. On the **Tables, Views, and Aliases (ADB223T)** panel, use the SEL line command to select the EMP table, and press Enter:



```
ADB223T n ----------- DD1A Tables, Views, and Aliases -------- Row 1 to 7 of 7
Command ===>                                                  Scroll ===> PAGE

Line commands:
 SEL - Select for SQL SELECT prototype  T - Table
 DEL - DELETE prototype  INS - INSERT prototype  UPD - UPDATE prototype

Select  Name               Schema   T
        *                  *        *
------- -----------------> -------- -
_______ AA1122             OWNER1   T
_______ AARVV1145600_ANDR  OWNER1   T
_______ EEMP               DSN8810  T
_______ EEPA               DSN8810  T
__SEL__ EMP                DSN8810  T
_______ EMPPROJACT         DSN8810  T
_______ EPROJ              DSN8810  T
_______ EPROJACT           DSN8810  T
_______ MAP_TBL            DSN8810  T
_______ NEWDEPT            DSN8810  T
_______ NEWPHONE           DSN8810  T
_______ PARTS              DSN8810  T
_______ PROJ               DSN8810  T
_______ PROJACT            DSN8810  T
_______ STAFF              DSN881SA T
_______ STAFFV1            DSN881SA V
_______ TCONA              DSN8810  T
_______ TDSPTXT            DSN8810  T
_______ TESTSTUFF          DSN881SA T
_______ TOPTVAL            DSN8810  T
_______ VACT               DSN8810  V
_______ VASTRDE1           DSN8810  V
_______ VASTRDE2           DSN8810  V
_______ VCONA              DSN8810  V
_______ VDEPMG1            DSN8810  V
_______ VDEPT              DSN8810  V
_______ VDSPTXT            DSN8810  V
_______ VEMP               DSN8810  V
```

The **Build SQL SELECT Prototype (ADB21TSE)** panel is displayed with a partially built SQL statement at the top.

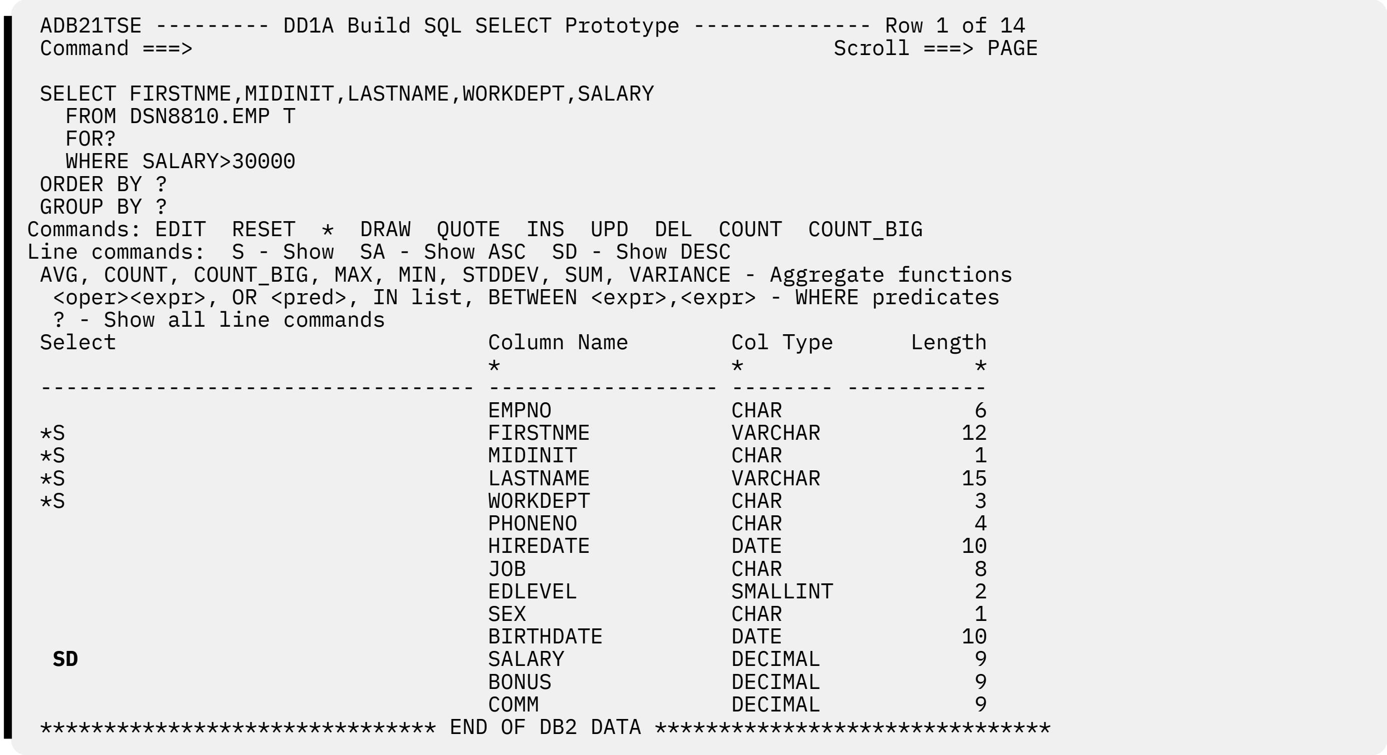
2. Use the S line command to include columns in your SELECT statement and the *<oper><expr>* line command to specify the salary range of greater than 30,000, and press Enter:



```
ADB21TSE --------- DD1A Build SQL SELECT Prototype -------------- Row 1 of 14
Command ===>                                                  Scroll ===> PAGE

SELECT ?
  FROM DSN8810.EMP T
  FOR?
  WHERE ?
ORDER BY ?
GROUP BY ?
Commands: EDIT  RESET  *  DRAW  QUOTE  INS  UPD  DEL  COUNT  COUNT_BIG
Line commands:  S - Show  SA - Show ASC  SD - Show DESC
 AVG, COUNT, COUNT_BIG, MAX, MIN, STDDEV, SUM, VARIANCE - Aggregate functions
 <oper><expr>, OR <pred>, IN list, BETWEEN <expr>,<expr> - WHERE predicates
 ? - Show all line commands

 Select                            Column Name        Col Type      Length
                                  *                  *                  *
---------------------------------- ------------------ -------- -----------
                                  EMPNO              CHAR               6
S                                 FIRSTNME           VARCHAR           12
S                                 MIDINIT            CHAR               1
S                                 LASTNAME           VARCHAR           15
S                                 WORKDEPT           CHAR               3
                                  PHONENO            CHAR               4
                                  HIREDATE           DATE              10
                                  JOB                CHAR               8
                                  EDLEVEL            SMALLINT           2
                                  SEX                CHAR               1
                                  BIRTHDATE          DATE              10
>30000                            SALARY             DECIMAL            9
                                  BONUS              DECIMAL            9
                                  COMM               DECIMAL            9
****************************** END OF DB2 DATA ******************************
```

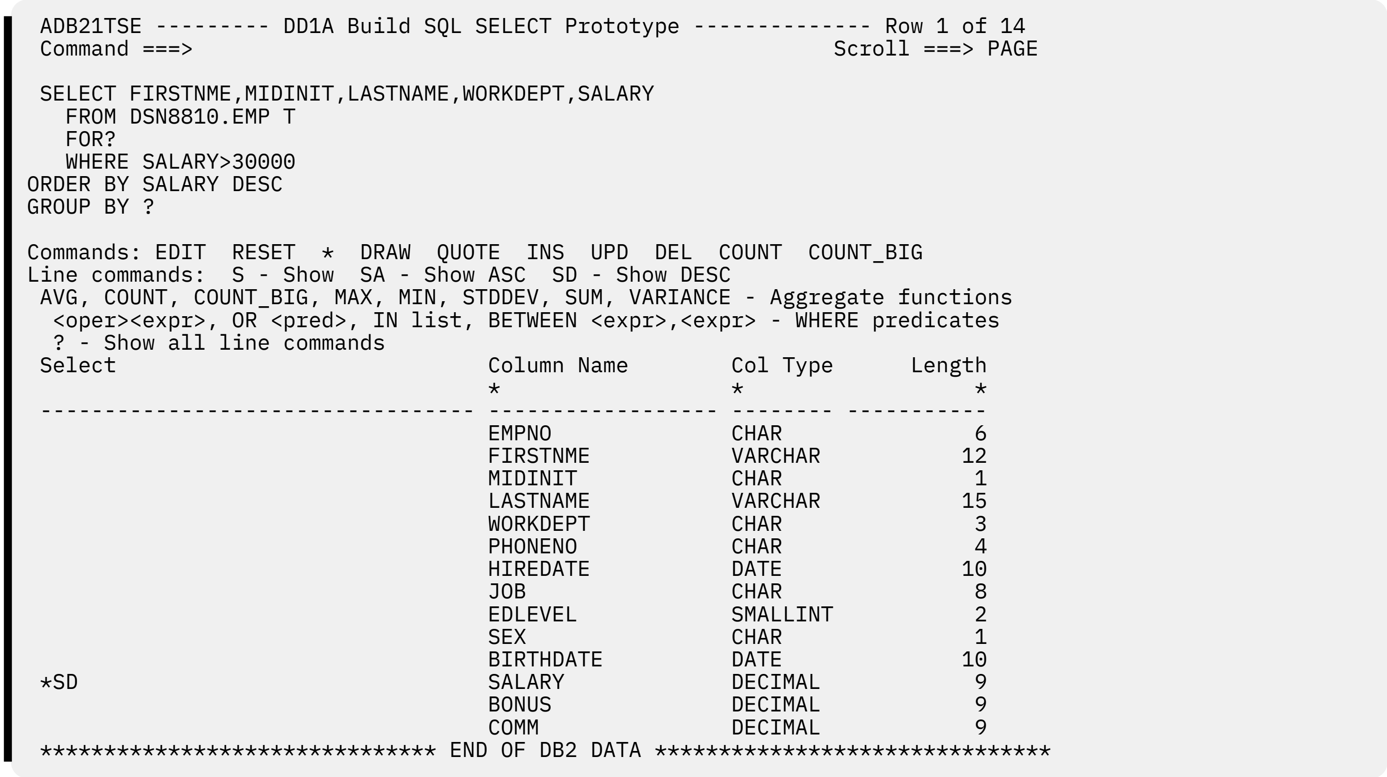
The SELECT statement is updated with these changes.

3. Use the SD line command to add the ORDER BY clause to the SELECT statement, and press Enter:



```
 ADB21TSE --------- DD1A Build SQL SELECT Prototype -------------- Row 1 of 14
 Command ===>                                                  Scroll ===> PAGE

 SELECT FIRSTNME,MIDINIT,LASTNAME,WORKDEPT,SALARY
   FROM DSN8810.EMP T
   FOR?
   WHERE SALARY>30000
 ORDER BY ?
 GROUP BY ?
Commands: EDIT  RESET  *  DRAW  QUOTE  INS  UPD  DEL  COUNT  COUNT_BIG
Line commands:  S - Show  SA - Show ASC  SD - Show DESC
 AVG, COUNT, COUNT_BIG, MAX, MIN, STDDEV, SUM, VARIANCE - Aggregate functions
  <oper><expr>, OR <pred>, IN list, BETWEEN <expr>,<expr> - WHERE predicates
  ? - Show all line commands
 Select                             Column Name        Col Type      Length
                                    *                  *                  *
 ---------------------------------- ------------------ -------- -----------
                                    EMPNO              CHAR               6
 *S                                 FIRSTNME           VARCHAR           12
 *S                                 MIDINIT            CHAR               1
 *S                                 LASTNAME           VARCHAR           15
 *S                                 WORKDEPT           CHAR               3
                                    PHONENO            CHAR               4
                                    HIREDATE           DATE              10
                                    JOB                CHAR               8
                                    EDLEVEL            SMALLINT           2
                                    SEX                CHAR               1
                                    BIRTHDATE          DATE              10
  SD                                SALARY             DECIMAL            9
                                    BONUS              DECIMAL            9
                                    COMM               DECIMAL            9
 ******************************* END OF DB2 DATA *******************************
```

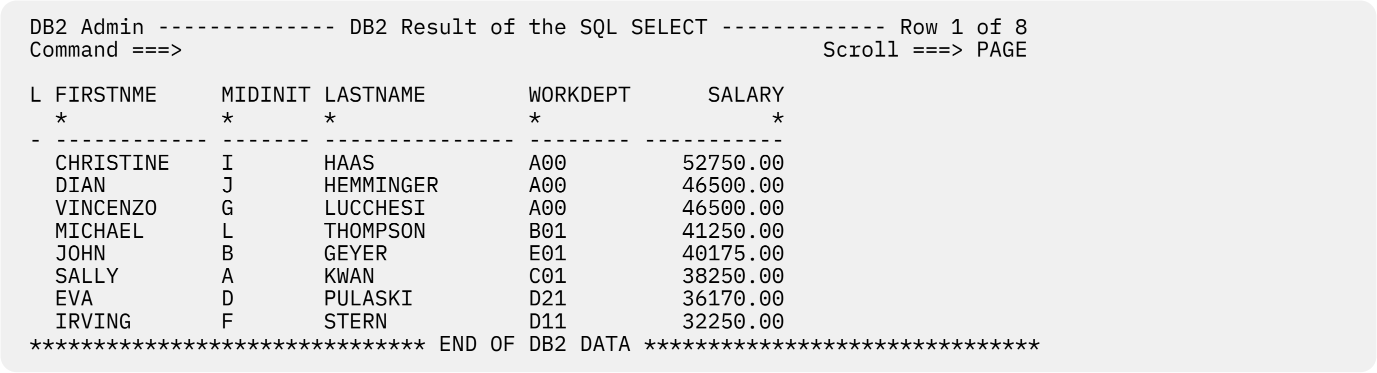
The SELECT statement is updated again as shown:



```
 ADB21TSE --------- DD1A Build SQL SELECT Prototype -------------- Row 1 of 14
 Command ===>                                                  Scroll ===> PAGE

 SELECT FIRSTNME,MIDINIT,LASTNAME,WORKDEPT,SALARY
   FROM DSN8810.EMP T
   FOR?
   WHERE SALARY>30000
ORDER BY SALARY DESC
GROUP BY ?

Commands: EDIT  RESET  *  DRAW  QUOTE  INS  UPD  DEL  COUNT  COUNT_BIG
Line commands:  S - Show  SA - Show ASC  SD - Show DESC
 AVG, COUNT, COUNT_BIG, MAX, MIN, STDDEV, SUM, VARIANCE - Aggregate functions
  <oper><expr>, OR <pred>, IN list, BETWEEN <expr>,<expr> - WHERE predicates
  ? - Show all line commands
 Select                             Column Name        Col Type      Length
                                    *                  *                  *
 ---------------------------------- ------------------ -------- -----------
                                    EMPNO              CHAR               6
                                    FIRSTNME           VARCHAR           12
                                    MIDINIT            CHAR               1
                                    LASTNAME           VARCHAR           15
                                    WORKDEPT           CHAR               3
                                    PHONENO            CHAR               4
                                    HIREDATE           DATE              10
                                    JOB                CHAR               8
                                    EDLEVEL            SMALLINT           2
                                    SEX                CHAR               1
                                    BIRTHDATE          DATE              10
 *SD                                SALARY             DECIMAL            9
                                    BONUS              DECIMAL            9
                                    COMM               DECIMAL            9
 ******************************* END OF DB2 DATA *******************************
```

The SQL statement is now ready to be run. Do not specify any line commands when running the statement.

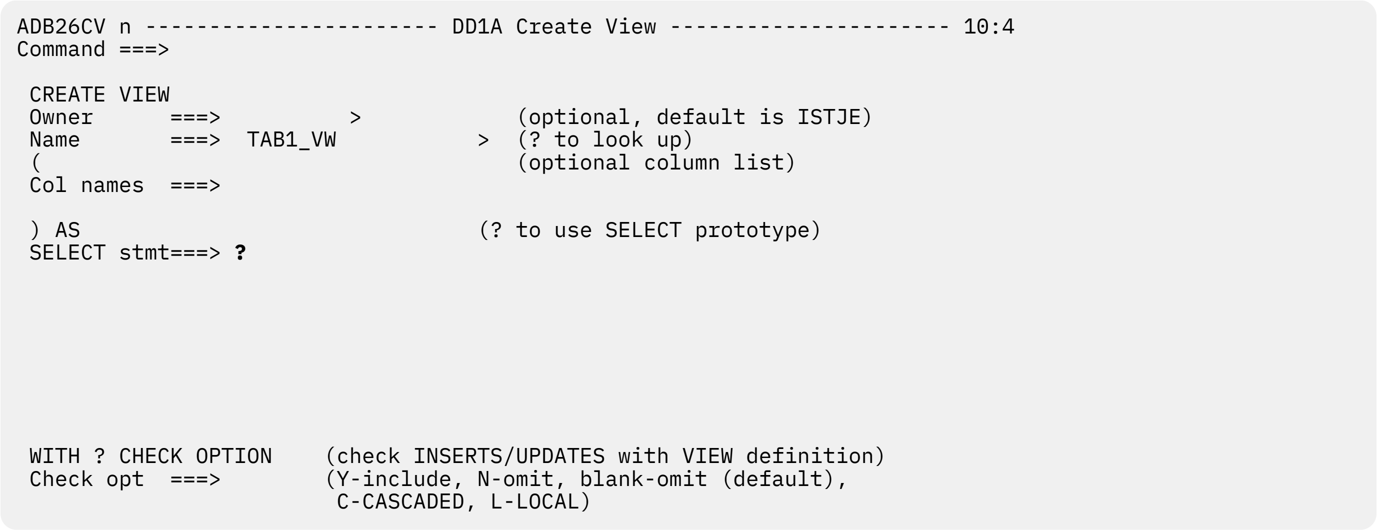
4. Press Enter to display the result of the SELECT statement:



```
DB2 Admin -------------- DB2 Result of the SQL SELECT ------------- Row 1 of 8
Command ===>                                                  Scroll ===> PAGE

L FIRSTNME     MIDINIT LASTNAME        WORKDEPT      SALARY
  *            *       *               *                  *
- ------------ ------- --------------- -------- -----------
  CHRISTINE    I       HAAS            A00         52750.00
  DIAN         J       HEMMINGER       A00         46500.00
  VINCENZO     G       LUCCHESI        A00         46500.00
  MICHAEL      L       THOMPSON        B01         41250.00
  JOHN         B       GEYER           E01         40175.00
  SALLY        A       KWAN            C01         38250.00
  EVA          D       PULASKI         D21         36170.00
  IRVING       F       STERN           D11         32250.00
******************************* END OF DB2 DATA *******************************
```

Example of building a SELECT statement when creating a view

You can also use prototyping to create a SELECT statement when creating a view. From the **Create View (ADB26CV)** panel, in the **SELECT stmt** field, enter ? to begin navigating to the **Build SQL SELECT Prototype (ADB21TSE)** panel where you can build a SELECT statement.



```
ADB26CV n ----------------------- DD1A Create View ---------------------- 10:4
Command ===>

 CREATE VIEW
 Owner      ===>          >            (optional, default is ISTJE)
 Name       ===>  TAB1_VW           >  (? to look up)
 (                                     (optional column list)
 Col names  ===>

 ) AS                               (? to use SELECT prototype)
 SELECT stmt===> ?








 WITH ? CHECK OPTION    (check INSERTS/UPDATES with VIEW definition)
 Check opt  ===>        (Y-include, N-omit, blank-omit (default),
                         C-CASCADED, L-LOCAL)
```

.

Build SQL Prototype panel

The **Build SQL Prototype** panel allows you to interactively build SELECT, INSERT, UPDATE, and DELETE statements.

Db2 Administration Tool has four versions of this panel:

Build SQL SELECT Prototype (ADB21TSE) panel
Build SQL INSERT Prototype (ADB21TIN) panel
Build SQL UPDATE Prototype (ADB21TUP) panel
Build SQL DELETE Prototype (ADB21TDE) panel

Primary commands

Depending on which version of the panel is displayed, one or more of the following primary commands are available:

*** (asterisk)**
Adds an asterisk to the query so that all columns are returned in the result.

COUNT(*)
Adds the COUNT(*) function to the query. This function returns the number of selected rows as an integer.

COUNT_BIG(*)
Adds the COUNT_BIG(*) function to the query. This function returns the number of selected rows as a decimal.

DEL
Opens the prototype panel to create a DELETE statement. DEL is not applicable to creating a view.

DRAW
Opens an ISPF editor with a query that includes all columns in the table. The data type for each column is included in comments. You can edit the statement from this editor.

EDIT
Opens an ISPF editor for you to edit the statement. Any changes that you make in the ISPF editor are not reflected in the **Build SQL Prototype** panel.

For SELECT statements, the difference between EDIT and DRAW is that EDIT does not automatically add all columns to the query.

FETCH
Adds FETCH FIRST ? ROWS ONLY to the DELETE statement. Replace the question mark (?) with an integer. This clause limits the number of rows that are deleted by the statement to the number that you specify.

INS
Opens the prototype panel to create a INSERT statement. INS is not applicable to creating a view.

QUOTE
Places quotation marks around column names.

RESET
Resets the query to how it looked when the **Build SQL Prototype** panel first opened. You can choose which clauses to reset (or to reset all of them) on a subsequent confirmation panel.

RUN
Executes the query.

SEL
Opens the prototype panel to create a SELECT statement.

UPD
Opens the prototype panel to create a UPDATE statement. UPD is not applicable to creating a view.

Line commands

Depending on which version of the panel is displayed, one or more of the following line commands are available:

S
Adds the column to the SELECT list so that is included (shown) in the result table.

SA
Adds the column to the SELECT list so that is included in the result table and indicates that you want the result set sorted in ascending order according to the values in this column. This column is added to the ORDER BY clause with the ASC keyword. SA is not applicable to creating a view.

SD
Adds the column to the SELECT list so that is included in the result table and indicates that you want the result set sorted in descending order according to the values in this column. This column is added to the ORDER BY clause with the DESC keyword. SD is not applicable to creating a view.

<value>
Inserts the specified value or expression into the column or updates the column with the specified value.

For example, if you are building an INSERT statement, on the **Build SQL INSERT Prototype (ADB21TIN)** panel, in the **Column values** column, enter the column values for the row that you want to insert:

```
Column values                       Column Name
                                    *
--------------------------------> ------------------
100                                 COL1
b                                   COL2
c                                   COL3
4000                                COL4
```

If you are building an UPDATE statement, on the **Build SQL UPDATE Prototype (ADB21TUP)** panel, in the **Expression or Predicate** column, enter the new value for the column:

```
                                    Column Name
 Expression or Predicate            *
 --------------------------------> ------------------
 200                                COL1
                                    COL2
                                    COL3
                                    COL4
```

The following line commands add the specified function to the query for the specified column:

AVG
Returns the average value for a numeric column.

COUNT
Returns the number of selected rows as an integer.

COUNT_BIG
Returns the number of selected rows as a decimal.

MAX
Returns the maximum value for a numeric column.

MIN
Returns the minimum value for a numeric column.

STDDEV
Returns the standard deviation for a numeric column.

SUM
Returns the sum of the selected columns.

VARIANCE
Returns the variance of a set of numbers from the selected columns.

The following line commands add predicates to the WHERE clause:

<oper><expr>
Adds a condition for the column.

<oper>
An operator. This value can be =, ¬=, >, >=, <, <=, <>, LIKE or IS.

<expr>
An expression. This value must be an alphanumeric.

<operator><value>
Adds a condition for the column.

<operator>
An operator. This value can be =, ¬=, >, >=, <, <=, <>, LIKE or IS.

<value>
Predicate, data value, or remainder of expression

Examples:

```
>=4
IS NOT NULL
```

OR *<pred>*

Adds the OR operator and a condition for the column.

<pred>

A predicate. This value can be an alphanumeric value or another valid operator, such as IN or BETWEEN.

Examples:

```
OR=10
OR=x
OR IN(1,2,3,4,5)
OR BETWEEN s,t
```

IN *list*

Adds the IN operator and possible values for the column.

list

A list of values.

Examples:

```
IN x,y
IN('x','y')
IN 1,2,3,4,5,6
```

BETWEEN *<expr>*, *<expr>*

Adds the BETWEEN operator and a range of values for the column. You can also specify BTW instead of BETWEEN.

<expr>

An expression

Examples:

```
BTW x,y
BETWEEN x AND y
BTW nnn,ppp
```

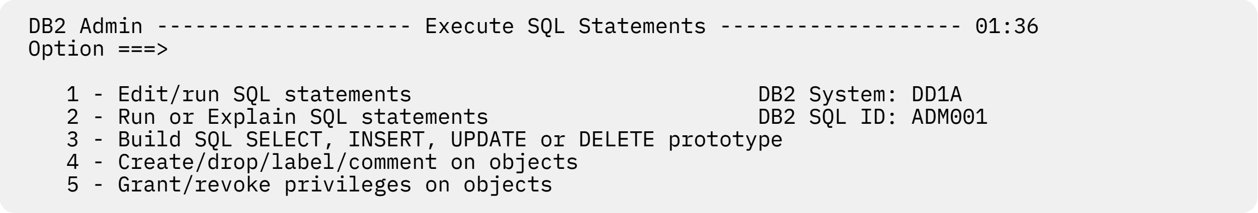
Issuing CREATE, DROP, LABEL ON, and COMMENT ON statements

Use the **Execute SQL Statements (ADB22)** panel to issue CREATE, DROP, LABEL ON, and COMMENT ON statements.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.

   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure:

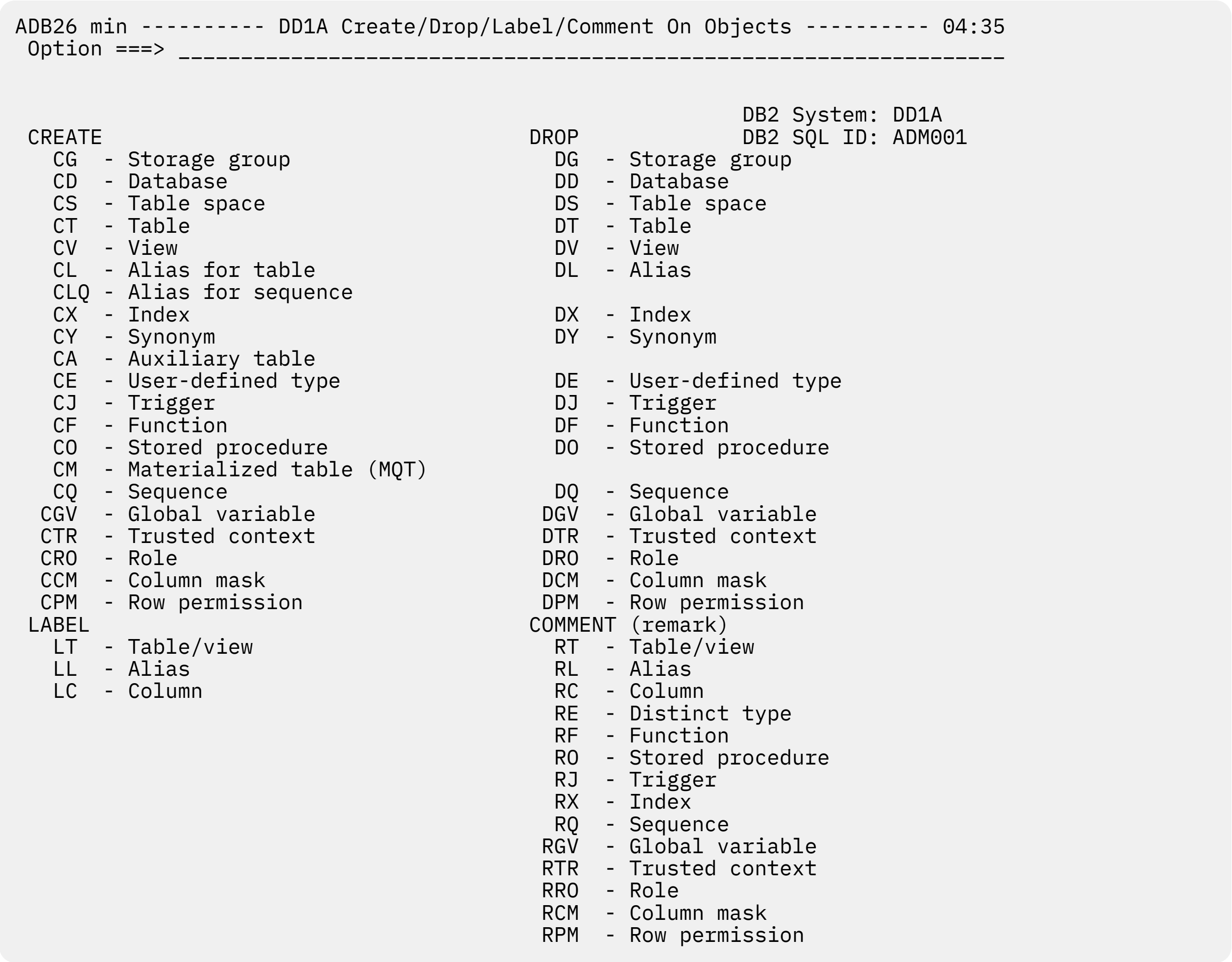


```
DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
Option ===>

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

*Figure 142. **Execute SQL Statements (ADB22)** panel*

2. Specify option 4, and press Enter.

   The **Create/Drop/Label/Comment On Objects (ADB26)** panel, as shown in the following figure:



```
ADB26 min ---------- DD1A Create/Drop/Label/Comment On Objects ---------- 04:35
 Option ===> __________________________________________________________________


                                                          DB2 System: DD1A
 CREATE                                  DROP             DB2 SQL ID: ADM001
   CG  - Storage group                     DG  - Storage group
   CD  - Database                          DD  - Database
   CS  - Table space                       DS  - Table space
   CT  - Table                             DT  - Table
   CV  - View                              DV  - View
   CL  - Alias for table                   DL  - Alias
   CLQ - Alias for sequence
   CX  - Index                             DX  - Index
   CY  - Synonym                           DY  - Synonym
   CA  - Auxiliary table
   CE  - User-defined type                 DE  - User-defined type
   CJ  - Trigger                           DJ  - Trigger
   CF  - Function                          DF  - Function
   CO  - Stored procedure                  DO  - Stored procedure
   CM  - Materialized table (MQT)
   CQ  - Sequence                          DQ  - Sequence
  CGV  - Global variable                  DGV  - Global variable
  CTR  - Trusted context                  DTR  - Trusted context
  CRO  - Role                             DRO  - Role
  CCM  - Column mask                      DCM  - Column mask
  CPM  - Row permission                   DPM  - Row permission
 LABEL                                   COMMENT (remark)
   LT  - Table/view                        RT  - Table/view
   LL  - Alias                             RL  - Alias
   LC  - Column                            RC  - Column
                                           RE  - Distinct type
                                           RF  - Function
                                           RO  - Stored procedure
                                           RJ  - Trigger
                                           RX  - Index
                                           RQ  - Sequence
                                          RGV  - Global variable
                                          RTR  - Trusted context
                                          RRO  - Role
                                          RCM  - Column mask
                                          RPM  - Row permission
```

Figure 143. ***Create/Drop/Label/Comment On Objects (ADB26)*** *panel*

What to do next

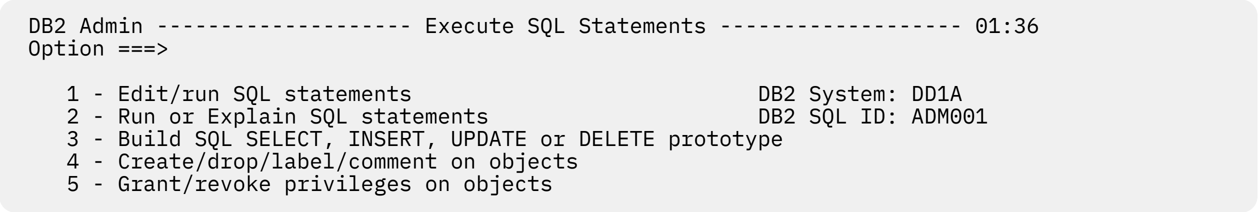
You can use the following examples as models when using panel ADB26 to create other objects.

Creating databases

Use the **Create Database (ADB26CD)** panel to create new databases.

Procedure

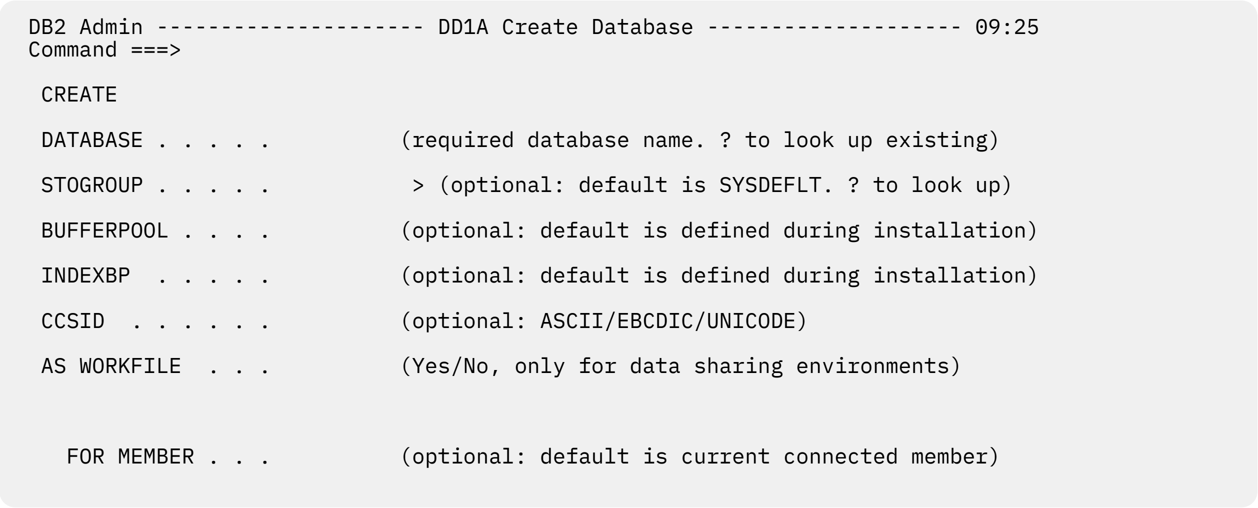
1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.
   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure:



```
DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
Option ===>

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

Figure 144. ***Execute SQL Statements (ADB22)*** *panel*

2. Specify option 4, and press Enter.
3. Specify option CD, and press Enter.
   The **Create Database (ADB26CD)** panel is displayed, as shown in the following figure.



```
DB2 Admin --------------------- DD1A Create Database -------------------- 09:25
Command ===>

 CREATE

 DATABASE . . . . .          (required database name. ? to look up existing)

 STOGROUP . . . . .           > (optional: default is SYSDEFLT. ? to look up)

 BUFFERPOOL . . . .          (optional: default is defined during installation)

 INDEXBP  . . . . .          (optional: default is defined during installation)

 CCSID  . . . . . .          (optional: ASCII/EBCDIC/UNICODE)

 AS WORKFILE  . . .          (Yes/No, only for data sharing environments)


   FOR MEMBER . . .          (optional: default is current connected member)
```

Figure 145. ***Create Database (ADB26CD)*** *panel*

4. Specify values for the following required parameters. Optionally, specify values for any of the other parameters:

   Required parameters

   DATABASE
   A database name for the new database, or enter a question mark (?) to look up existing database names using **Databases (ADB21D)** panel.

   Optional parameters

   STOGROUP
   The name of a storage group in which you want the new database to belong.

   BUFFERPOOL
   INDEXBP
   The names of buffer pools to use as defined at installation time.

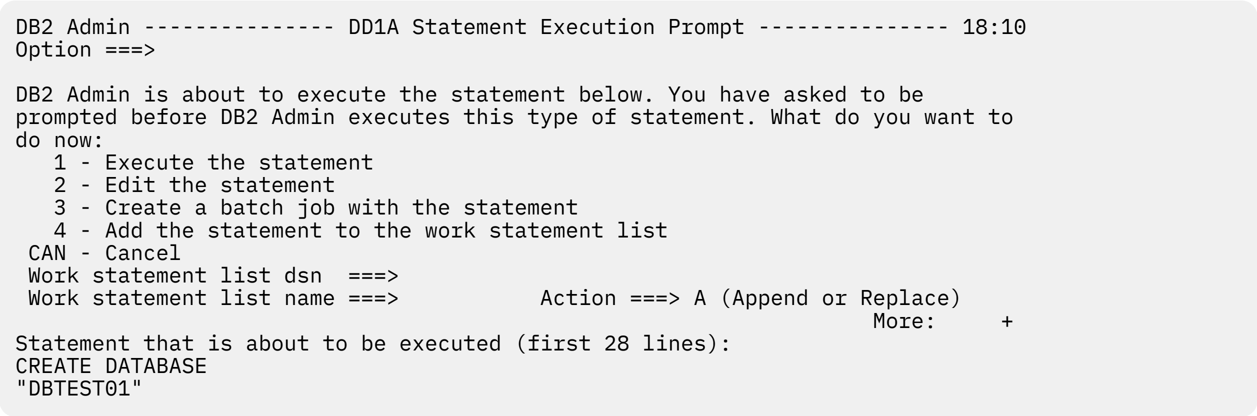
   CCSID
   One of the following formats: `ASCII`, `EBCDIC`, or `UNICODE`.

   AS WORKFILE
   The work file name for data sharing environments.

   AS TEMP
   Indicates whether to create a database for declared temporary tables.

   FOR MEMBER
   A different member in which to place the new database.

5. Follow the instructions on the **Statement Execution Prompt (ADB2PSTM)** panel panel (if enabled), as shown in the following figure, to complete and run the SQL statement for creating the new database.



```
DB2 Admin --------------- DD1A Statement Execution Prompt --------------- 18:10
Option ===>

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
   1 - Execute the statement
   2 - Edit the statement
   3 - Create a batch job with the statement
   4 - Add the statement to the work statement list
 CAN - Cancel
 Work statement list dsn  ===>
 Work statement list name ===>           Action ===> A (Append or Replace)
                                                                   More:     +
Statement that is about to be executed (first 28 lines):
CREATE DATABASE
"DBTEST01"
```

*Figure 146. The **Statement Execution Prompt (ADB2PSTM)** panel) – Creating a new database*

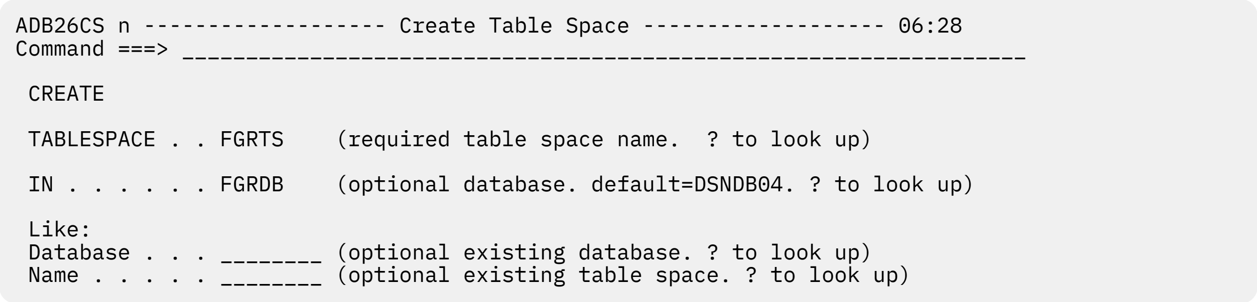
Creating table spaces

You can use Db2 Admin Tool to create partition-by-growth (PBG) and partition-by-range (PBR) table spaces.

Procedure

To create a table space:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.
2. On the **Execute SQL Statements (ADB22)** panel, specify option 4, and press Enter.
3. On the **Create/Drop/Label/Comment On Objects (ADB26)** panel, specify option CS, and press Enter.
4. On the **Create Table Space (ADB26CS)** panel, specify the table space name and any other parameters as needed, and press Enter:



```
ADB26CS n ------------------- Create Table Space ------------------- 06:28
Command ===> __________________________________________________________________

 CREATE

 TABLESPACE . . FGRTS    (required table space name.  ? to look up)

 IN . . . . . . FGRDB    (optional database. default=DSNDB04. ? to look up)

 Like:
 Database . . . ________ (optional existing database. ? to look up)
 Name . . . . . ________ (optional existing table space. ? to look up)
```

Required parameters:

TABLESPACE
A name for the new table space.

Optional parameters:

IN
The name of the database in which you want to create the new table space. The default is DSNDB04.

Like:
Database
Name
The table space (and containing database) on which you want to model the new table space.

Enter a question mark (?) to look up the names of existing objects for any of these fields.

5. On the **Create Table Space (ADB21SAR)** panel, set up a partitioning method for the table space by issuing one of the following commands:

MAKEPBG
Initializes the input fields as follows for a partition-by-growth table space (PBG) :

Numparts=0
Max Partitions >0

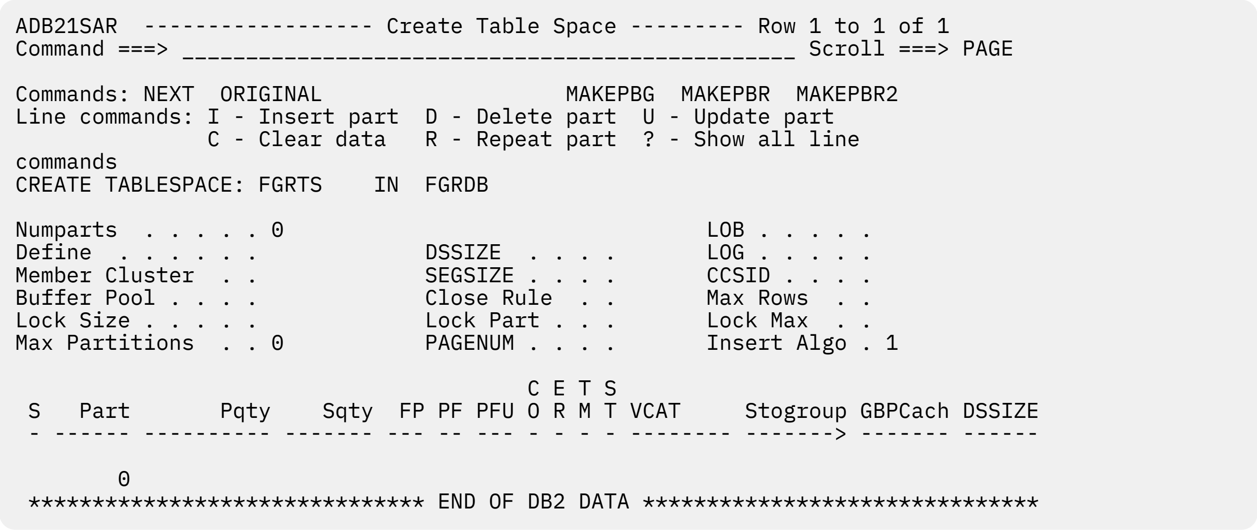
MAKEPBR
Initializes the input fields as follows for a partition-by-range table space (PBR) :

Numparts>0
Max Partitions=0
SEGSIZE >0

MAKEPBR2
Initializes the input fields as follows for a partition-by-range table space with relative page numbering (PBR RPN or PBR2):

Numparts>0
Max Partitions=0
SEGSIZE >0
PAGENUM=R

6. Specify any other parameters for the new table space and issue the NEXT command.



```
ADB21SAR  ------------------ Create Table Space --------- Row 1 to 1 of 1
Command ===> ________________________________________________ Scroll ===> PAGE

Commands: NEXT  ORIGINAL                   MAKEPBG  MAKEPBR  MAKEPBR2
Line commands: I - Insert part  D - Delete part  U - Update part
               C - Clear data   R - Repeat part  ? - Show all line
commands
CREATE TABLESPACE: FGRTS    IN  FGRDB

Numparts  . . . . . 0                            LOB . . . . .
Define  . . . . . .             DSSIZE  . . . .       LOG . . . . .
Member Cluster  . .             SEGSIZE . . . .       CCSID . . . .
Buffer Pool . . . .             Close Rule  . .       Max Rows  . .
Lock Size . . . . .             Lock Part . . .       Lock Max  . .
Max Partitions  . . 0           PAGENUM . . . .       Insert Algo . 1

                                        C E T S
 S   Part       Pqty    Sqty  FP PF PFU O R M T VCAT     Stogroup GBPCach DSSIZE
 - ------ ---------- ------- --- -- --- - - - - -------- -------> ------- ------

        0
 ******************************* END OF DB2 DATA *******************************
```

Note: Insert Algo is displayed only if you are running Db2 12 for z/OS.

If the statement execution prompt is not enabled, a CREATE TABLESPACE statement is executed with the parameters that you specified, and the table space is created.

7. If the **Statement Execution Prompt (ADB2PSTM)** panel is displayed, follow the instructions on that panel to complete and run the SQL statement to create the table space.

Creating tables

When you create a table in Db2 Admin Tool, you define all of the table attributes and columns. To save time, you can specify that you want to use another table as a model.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.
2. On the **Execute SQL Statements (ADB22)** panel, specify option 4, and press Enter:

```
ADB22 min -------------------- Execute SQL Statements ------------------- 01:36
Option ===> 4

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

3. On the **Create/Drop/Label/Comment On Objects (ADB26)** panel, specify option CT, and press Enter.
4. On the **Create Table (ADB26CT)** panel, specify values for the following required parameters and any optional parameters, and press Enter:

Required parameters

Schema
Specify the schema for the new table or use the default schema.

Name
Specify a table name for the new table. Enter a question mark (?) to look up existing table names on the **Tables, Views, and Aliases (ADB21T)** panel.

Number of columns
Specify the number of columns for the new table.

Optional parameters

LIKE
Specify these parameters if you want to create your new table like an existing table.

Schema
Specify the schema for the existing table.

Name
Specify the name of the existing table.

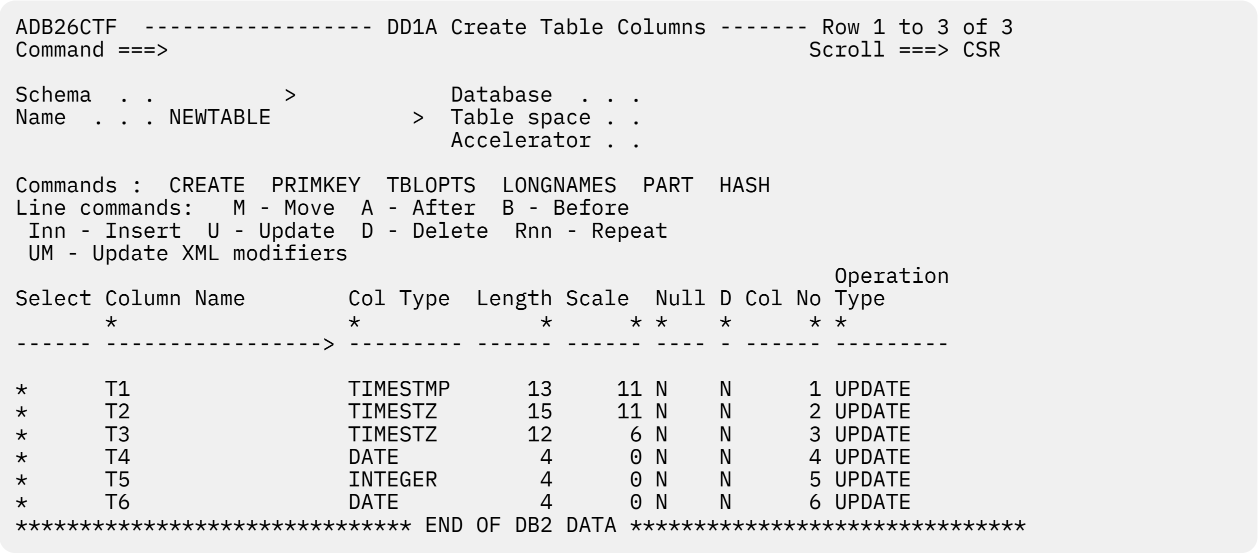
Identity attrs
Specify whether to include identity column attributes in the new table.

Row chg attrs
Specify whether to include row change timestamp column attributes in the new table.

As model only
Specify whether to use the existing table as a model and edit its columns values.

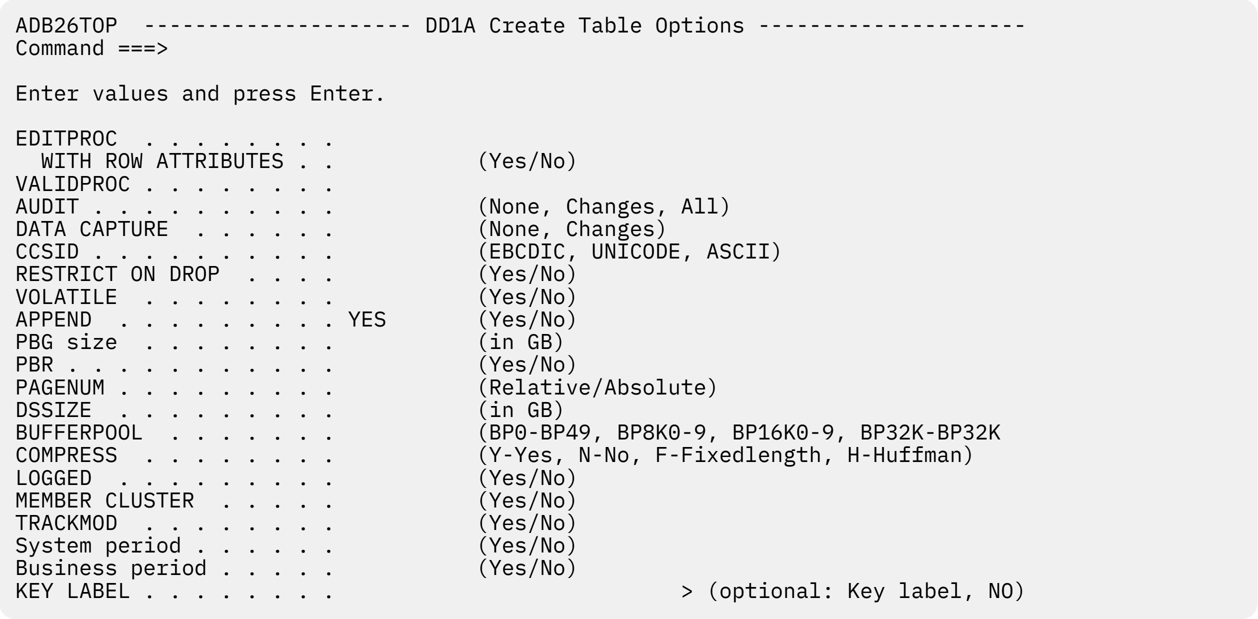
```
ADB26CT n ----------------------DD1A Create Table ---------------------- 16:31
Command ===>

 CREATE TABLE

 Schema  . . . . .           >           (default is )
 Name  . . . . . . NEW                    > (? to look up)

 LIKE
 Schema  . . . . .           >
 Name  . . . . . .                     > (? to look up)
 Identity attrs  .                   (Yes/No)
 Row chg attrs . .                   (Yes/No)
 As model only . .                   (Yes/No)

 (
 Number of columns . . . 6
```

5. On the **Create Table Columns (ADB26CTF)** panel, specify columns and column attributes for the new table:



```
ADB26CTF  ------------------ DD1A Create Table Columns ------- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Schema  . .          >            Database  . . .
Name  . . . NEWTABLE           >  Table space . .
                                  Accelerator . .

Commands :  CREATE  PRIMKEY  TBLOPTS  LONGNAMES  PART  HASH
Line commands:   M - Move  A - After  B - Before
 Inn - Insert  U - Update  D - Delete  Rnn - Repeat
 UM - Update XML modifiers
                                                                Operation
Select Column Name        Col Type  Length Scale  Null D Col No Type
       *                  *              *      * *    *      * *
------ ----------------> --------- ------ ------ ---- - ------ ---------

*      T1                 TIMESTMP      13     11 N    N      1 UPDATE
*      T2                 TIMESTZ       15     11 N    N      2 UPDATE
*      T3                 TIMESTZ       12      6 N    N      3 UPDATE
*      T4                 DATE           4      0 N    N      4 UPDATE
*      T5                 INTEGER        4      0 N    N      5 UPDATE
*      T6                 DATE           4      0 N    N      6 UPDATE
****************************** END OF DB2 DATA ******************************
```

6. Optional: If you want to specify additional table attributes, issue the TBLOPTS command and press Enter. Then, on the **Create Table Options (ADB26TOP)** panel, specify the appropriate values, and press Enter:



```
ADB26TOP  -------------------- DD1A Create Table Options --------------------
Command ===>

Enter values and press Enter.

EDITPROC  . . . . . . . .
  WITH ROW ATTRIBUTES . .           (Yes/No)
VALIDPROC . . . . . . . .
AUDIT . . . . . . . . . .           (None, Changes, All)
DATA CAPTURE  . . . . . .           (None, Changes)
CCSID . . . . . . . . . .           (EBCDIC, UNICODE, ASCII)
RESTRICT ON DROP  . . . .           (Yes/No)
VOLATILE  . . . . . . . .           (Yes/No)
APPEND  . . . . . . . . . YES       (Yes/No)
PBG size  . . . . . . . .           (in GB)
PBR . . . . . . . . . . .           (Yes/No)
PAGENUM . . . . . . . . .           (Relative/Absolute)
DSSIZE  . . . . . . . . .           (in GB)
BUFFERPOOL  . . . . . . .           (BP0-BP49, BP8K0-9, BP16K0-9, BP32K-BP32K
COMPRESS  . . . . . . . .           (Y-Yes, N-No, F-Fixedlength, H-Huffman)
LOGGED  . . . . . . . . .           (Yes/No)
MEMBER CLUSTER  . . . . .           (Yes/No)
TRACKMOD  . . . . . . . .           (Yes/No)
System period . . . . . .           (Yes/No)
Business period . . . . .           (Yes/No)
KEY LABEL . . . . . . . .                           > (optional: Key label, NO)
```

7. On the **Create Table Columns (ADB26CTF)** panel, issue the CREATE command, and press Enter.
8. If the **Statement Execution Prompt (ADB2PSTM)** panel is displayed, follow the instructions on that panel to complete and run the SQL statement.

Results

The CREATE TABLE statement is executed and the table is created.

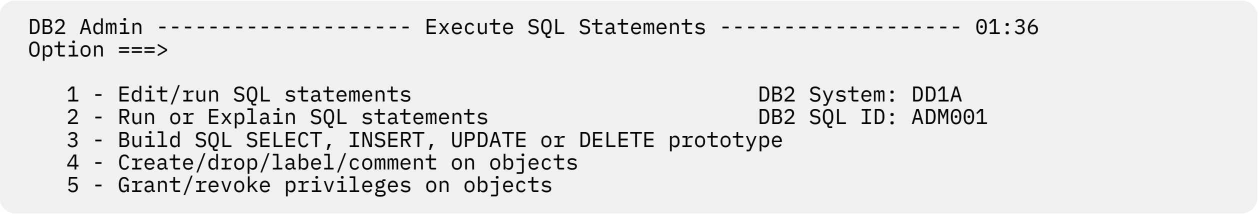
Creating materialized query tables

Use the **Create Materialized Table (ADB26CM)** panel to create a new materialized query table.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.

   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure:

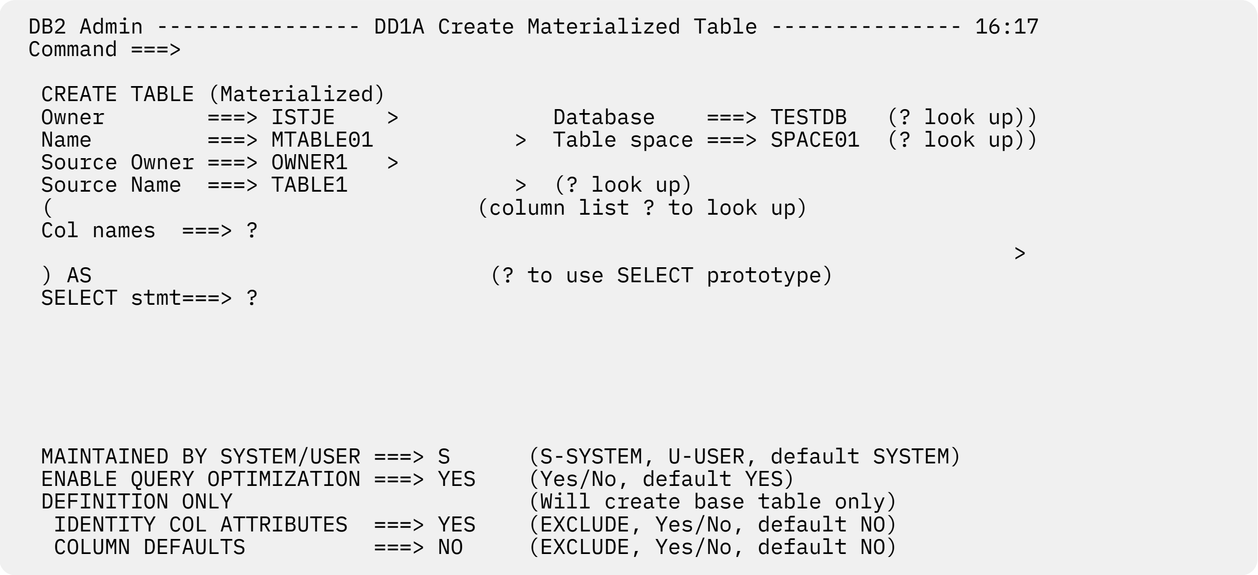


```
DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
Option ===>

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

Figure 147. ***Execute SQL Statements (ADB22)*** *panel*

2. Specify option 4, and press Enter.
3. Specify option CM, and press Enter.

   The **Create Materialized Table (ADB26CM)** panel is displayed, as shown in the following figure.



```
DB2 Admin ---------------- DD1A Create Materialized Table --------------- 16:17
Command ===>

 CREATE TABLE (Materialized)
 Owner        ===> ISTJE    >            Database    ===> TESTDB   (? look up))
 Name         ===> MTABLE01           >  Table space ===> SPACE01  (? look up))
 Source Owner ===> OWNER1   >
 Source Name  ===> TABLE1             >  (? look up)
 (                                 (column list ? to look up)
 Col names  ===> ?
                                                                            >
 ) AS                               (? to use SELECT prototype)
 SELECT stmt===> ?






 MAINTAINED BY SYSTEM/USER ===> S      (S-SYSTEM, U-USER, default SYSTEM)
 ENABLE QUERY OPTIMIZATION ===> YES    (Yes/No, default YES)
 DEFINITION ONLY                       (Will create base table only)
  IDENTITY COL ATTRIBUTES  ===> YES    (EXCLUDE, Yes/No, default NO)
  COLUMN DEFAULTS          ===> NO     (EXCLUDE, Yes/No, default NO)
```

Figure 148. ***Create Materialized Table (ADB26CM)*** *panel*

4. Specify values for the following parameters, and press Enter.

   Owner
   A table owner name for the new materialized query table.

   Name
   A table name for the new materialized query table.

   Source Owner
   The name of the owner of that source table.

   Col names
   The column names to be added to the new materialized query table, or enter a question mark (?) to look up existing column names.

   SELECT stmt
   An SQL SELECT statement to build the materialized query table, or enter a question mark (?) to use the Build SQL SELECT Prototype panel to build one.

   MAINTAINED BY SYSTEM/USER
   Indicates whether you want the Db2 system (S) or the user (U) to update and maintain the table.

   ENABLE QUERY OPTIMIZATION
   Indicates whether to use the Db2 query optimizer.

   IDENTITY COL ATTRIBUTES
   Indicates whether to exclude identity column attributes. Specify Yes to exclude them. Otherwise, specify No.

COLUMN DEFAULTS
Indicates whether to exclude column defaults. Specify Yes to exclude them. Otherwise, specify No.

If you specify Yes for both fields in the DEFINITION ONLY area, Db2 Admin Tool creates a regular base table (type "T"), as opposed to a materialized query table.

5. Follow the instructions on the **Statement Execution Prompt (ADB2PSTM)** panel panel (if enabled) to complete and run the SQL statement for creating the new materialized query table.

Creating indexes on tables

Use the **Create Index (ADB26CX)** panel to create new indexes on tables.

About this task

Using Db2 Admin Tool, you can create a new index on a table by using several methods:

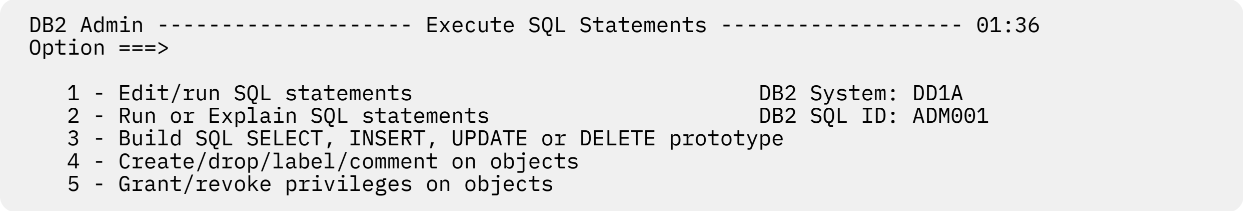
- Select option CX on the **Create/Drop/Label/Comment On Objects (ADB26)** panel. The following procedure explains how to use this method.
- Use the CREX line command on the **Tables, Views, and Aliases (ADB21T)** panel.
- Use the CRE line command on the **Indexes (ADB21X)** panel.
- Select option CX on the **EXPLAIN (ADB2E)** panel.

Each of these methods display the create index panels, beginning with the **Create Index (ADB26CX)** panel.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.

   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure:

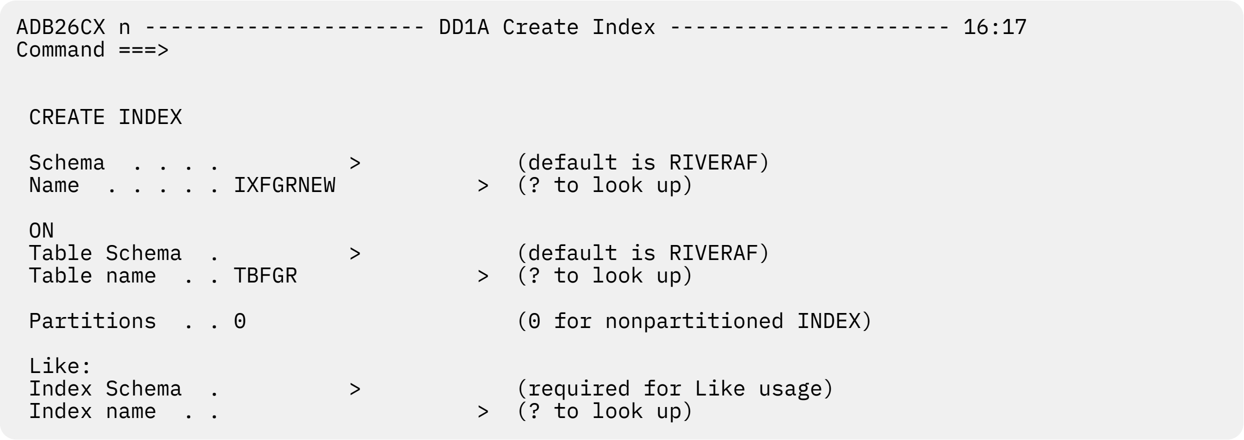


```
DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
Option ===>

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

*Figure 149. **Execute SQL Statements (ADB22)** panel*

2. Specify option 4, and press Enter.
3. Specify option CX, and press Enter.

   The **Create Index (ADB26CX)** panel is displayed, as shown in the following figure.



```
ADB26CX n --------------------- DD1A Create Index --------------------- 16:17
Command ===>


 CREATE INDEX

 Schema  . . . .          >            (default is RIVERAF)
 Name  . . . . . IXFGRNEW           >  (? to look up)

 ON
 Table Schema  .          >            (default is RIVERAF)
 Table name  . . TBFGR              >  (? to look up)

 Partitions  . . 0                     (0 for nonpartitioned INDEX)

 Like:
 Index Schema  .          >            (required for Like usage)
 Index name  . .                    >  (? to look up)
```

Figure 150. ***Create Index (ADB26CX)*** *panel*

4. Specify values for the required parameters and for any of the other parameters, and press Enter.

   Required parameters

   Owner
   The name of the index owner for the new index or use the default owner.

   Name
   An index name for the new index, or enter a question mark (?) to look up existing index names using the **Indexes (ADB21X)** panel.

   Table name
   A table name or enter a question mark (?) to look up existing table names using the **Tables, Views, and Aliases (ADB21T)** panel.

   Optional parameters

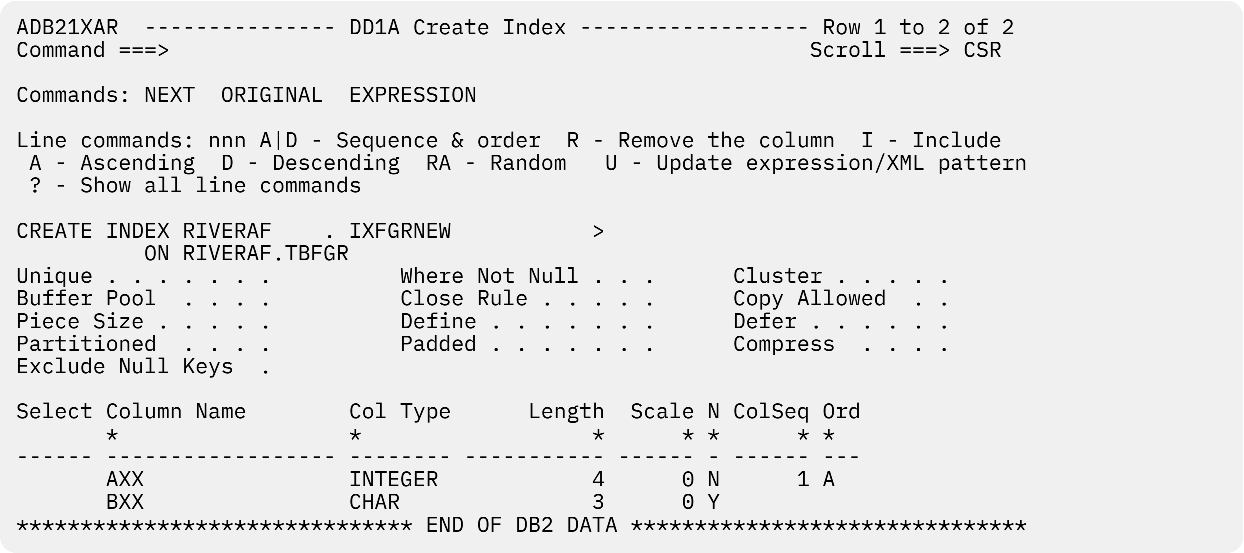
   Table owner
   The name of the owner for a table name on which the index is based.

   Partitions
   The number of partitions for a partitioned index.

   Like: Index owner
   The name of an owner on which to model the new owner for the new index.

   Like: Index name
   The name of an index on which to model the new index or enter a question mark (?) to look up existing index names using the **Indexes (ADB21X)** panel.

   When you press Enter, the next **Create Index (ADB21XAR)** panel is displayed, as shown in the following figure.



```
ADB21XAR  --------------- DD1A Create Index ----------------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> CSR

Commands: NEXT  ORIGINAL  EXPRESSION

Line commands: nnn A|D - Sequence & order  R - Remove the column  I - Include
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern
 ? - Show all line commands

CREATE INDEX RIVERAF    . IXFGRNEW           >
          ON RIVERAF.TBFGR
Unique . . . . . . .          Where Not Null . . .      Cluster . . . . .
Buffer Pool  . . . .          Close Rule . . . . .      Copy Allowed  . .
Piece Size . . . . .          Define . . . . . . .      Defer . . . . . .
Partitioned  . . . .          Padded . . . . . . .      Compress  . . . .
Exclude Null Keys  .

Select Column Name        Col Type      Length  Scale N ColSeq Ord
       *                  *                  *      * *      * *
------ ------------------ --------  ---------- ------ - ------ ---
       AXX                INTEGER            4      0 N      1 A
       BXX                CHAR               3      0 Y
******************************* END OF DB2 DATA *******************************
```

*Figure 151. **Create Index (ADB21XAR)** panel*

5. On the upper portion of the **Create Index (ADB21XAR)** panel, specify the index attributes.
   a) Specify whether the columns should be in ascending or descending order.
   b) Specify the general index attributes in the fields.

      Remember: Depending upon the version of Db2 that you are using and your choice of parameters, some of the attribute fields might be unavailable.

      Tip: Without negatively impacting query performance, you can improve the insert performance of NULL entries, by excluding NULL rows from an index. Type `Yes` in the **Exclude Null Keys** field to exclude NULL rows from a new index. The default is to include NULL keys in a new index.
6. On the scrollable table of the panel, use line commands to specify the columns in the index. All columns of the Db2 table are displayed. Index columns are identified in the Colseq and Order columns.
7. Issue the NEXT primary command to display the **Create Index - Space (ADB21XAS)** panel.
8. Specify the space allocation and storage parameters for the index or for each partition of the index.

   If a partitioned index with more than one partition is being created, the word `Default:` appears at the beginning of the scrollable portion of the panel. Use this line to enter common values for each partition and to avoid entering the same value for a parameter on all partitions again.
9. Issue the NEXT primary command to generate the DDL for the index and display an edit session.
10. Edit the CREATE statement or exit the session to create the index.
11. Follow the instructions on the **Statement Execution Prompt (ADB2PSTM)** panel (if enabled) to complete and run the SQL statement for creating the new index.

Results

After the index is created, Db2 Admin Tool displays the **Create Index Utilities (ADB26CXU)** panel, on which you can run several index utilities, including RECOVER and RUNSTATS.

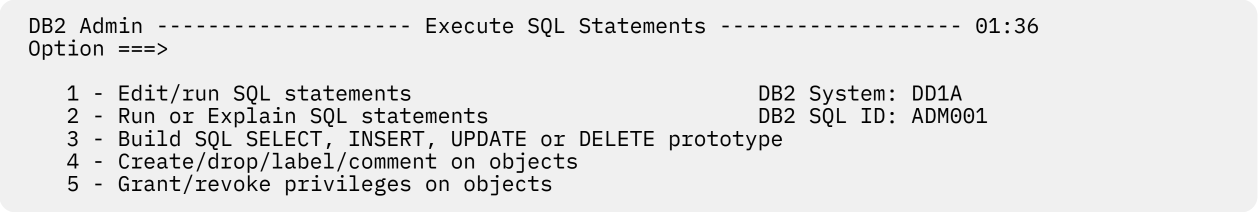
Creating triggers

Use the **Create Trigger (ADB26CJ)** panel to create basic and advanced triggers.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.

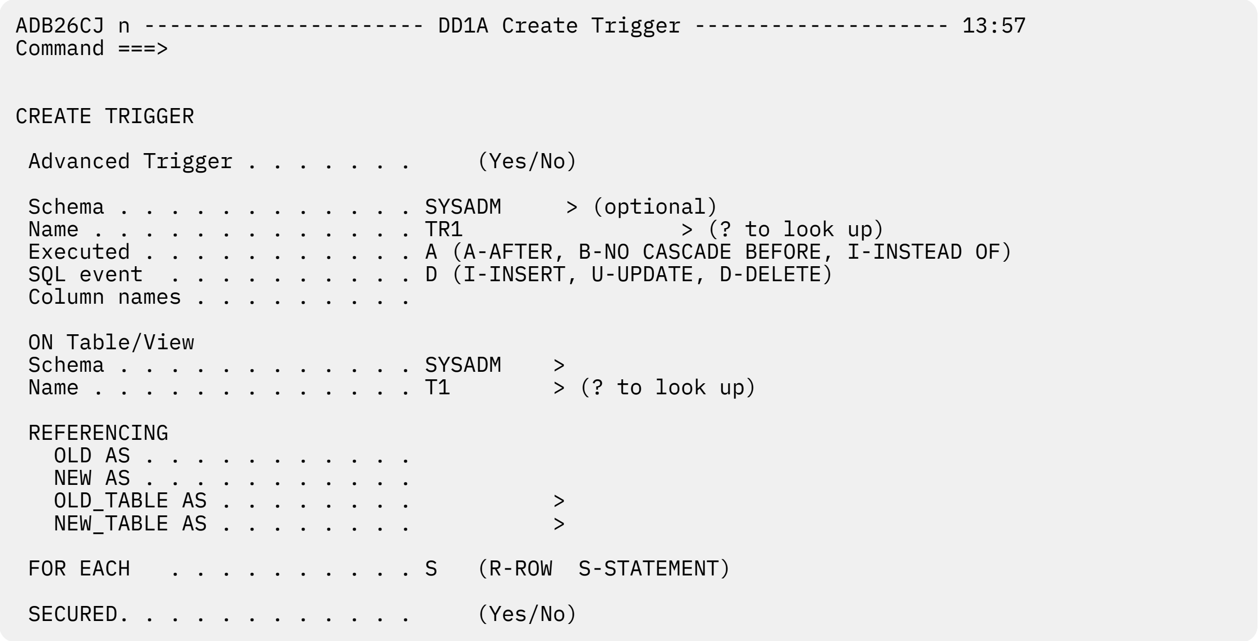
   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure:



```
DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
Option ===>

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

Figure 152. ***Execute SQL Statements (ADB22)*** *panel*

2. Specify option 4, and press Enter.
3. Specify option CJ, and press Enter.
4. Specify option CRE on the **Triggers (ADB21J)** panel. The **Create Trigger (ADB26CJ)** panel is displayed, as shown in the following figure.



```
ADB26CJ n --------------------- DD1A Create Trigger -------------------- 13:57
Command ===>


CREATE TRIGGER

 Advanced Trigger . . . . . . .     (Yes/No)

 Schema . . . . . . . . . . . . SYSADM     > (optional)
 Name . . . . . . . . . . . . . TR1                > (? to look up)
 Executed . . . . . . . . . . . A (A-AFTER, B-NO CASCADE BEFORE, I-INSTEAD OF)
 SQL event  . . . . . . . . . . D (I-INSERT, U-UPDATE, D-DELETE)
 Column names . . . . . . . . .

 ON Table/View
 Schema . . . . . . . . . . . . SYSADM    >
 Name . . . . . . . . . . . . . T1        > (? to look up)

 REFERENCING
   OLD AS . . . . . . . . . . .
   NEW AS . . . . . . . . . . .
   OLD_TABLE AS . . . . . . . .           >
   NEW_TABLE AS . . . . . . . .           >

 FOR EACH   . . . . . . . . . . S   (R-ROW  S-STATEMENT)

 SECURED. . . . . . . . . . . .     (Yes/No)
```

Figure 153. ***Create Trigger (ADB26CJ)*** *panel*

5. Specify values for the following parameters, and press Enter.

CREATE TRIGGER
The CREATE TRIGGER section contains the following parameters:

Advanced trigger
Specify Yes to create an advanced trigger. Otherwise, specify No.

Schema
The schema. This field is optional.

Name
The name of the trigger.

Executed
Indicates when the trigger is executed. Valid values are A, B, and I.

SQL event
Indicates the triggering event. Valid values are I, U, and D.

ON Table/View
The ON Table/View section contains the following parameters:

Schema
The schema. This field is optional.

Name
The name of the table or view.

REFERENCING
The REFERENCING section contains the following parameters. For these, specify the correlation names for the transition variables and the table names for the transition tables.

OLD AS
The correlation name that identifies the values in the row prior to the triggering SQL operation.

NEW AS
The correlation name that identifies the values in the row as modified by the triggering SQL operation and by any assignment statement in a before trigger that has already been executed.

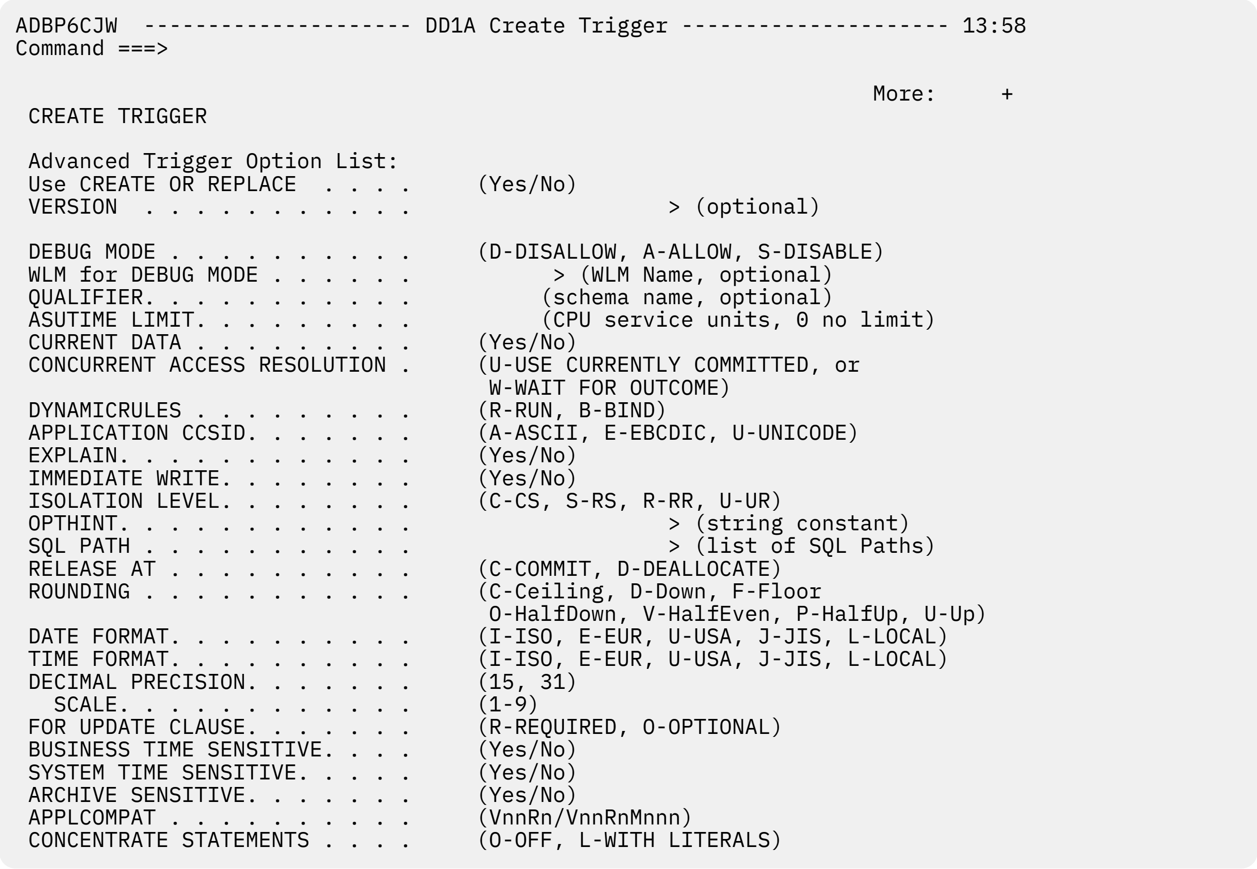
OLD TABLE AS
The name of a temporary table that identifies the values in the complete set of rows that are modified rows by the triggering SQL operation prior to any actual changes.

NEW TABLE AS
The name of a temporary table that identifies the values in the complete set of rows as modified by the triggering SQL operation and by any assignment statement in a before trigger that has already been executed.

FOR EACH
Indicates trigger granularity. Valid values are R and S.

SECURED
Indicates whether the trigger is secured.

If you specified that you want to create an advanced trigger, the **Advanced trigger options (ADBP6CJW)** panel is displayed, as shown in the following figure:



```
ADBP6CJW  --------------------- DD1A Create Trigger --------------------- 13:58
Command ===>

                                                                   More:     +
 CREATE TRIGGER

 Advanced Trigger Option List:
 Use CREATE OR REPLACE  . . . .     (Yes/No)
 VERSION  . . . . . . . . . . .                    > (optional)

 DEBUG MODE . . . . . . . . . .     (D-DISALLOW, A-ALLOW, S-DISABLE)
 WLM for DEBUG MODE . . . . . .           > (WLM Name, optional)
 QUALIFIER. . . . . . . . . . .          (schema name, optional)
 ASUTIME LIMIT. . . . . . . . .          (CPU service units, 0 no limit)
 CURRENT DATA . . . . . . . . .     (Yes/No)
 CONCURRENT ACCESS RESOLUTION .     (U-USE CURRENTLY COMMITTED, or
                                     W-WAIT FOR OUTCOME)
 DYNAMICRULES . . . . . . . . .     (R-RUN, B-BIND)
 APPLICATION CCSID. . . . . . .     (A-ASCII, E-EBCDIC, U-UNICODE)
 EXPLAIN. . . . . . . . . . . .     (Yes/No)
 IMMEDIATE WRITE. . . . . . . .     (Yes/No)
 ISOLATION LEVEL. . . . . . . .     (C-CS, S-RS, R-RR, U-UR)
 OPTHINT. . . . . . . . . . . .                    > (string constant)
 SQL PATH . . . . . . . . . . .                    > (list of SQL Paths)
 RELEASE AT . . . . . . . . . .     (C-COMMIT, D-DEALLOCATE)
 ROUNDING . . . . . . . . . . .     (C-Ceiling, D-Down, F-Floor
                                     O-HalfDown, V-HalfEven, P-HalfUp, U-Up)
 DATE FORMAT. . . . . . . . . .     (I-ISO, E-EUR, U-USA, J-JIS, L-LOCAL)
 TIME FORMAT. . . . . . . . . .     (I-ISO, E-EUR, U-USA, J-JIS, L-LOCAL)
 DECIMAL PRECISION. . . . . . .     (15, 31)
   SCALE. . . . . . . . . . . .     (1-9)
 FOR UPDATE CLAUSE. . . . . . .     (R-REQUIRED, O-OPTIONAL)
 BUSINESS TIME SENSITIVE. . . .     (Yes/No)
 SYSTEM TIME SENSITIVE. . . . .     (Yes/No)
 ARCHIVE SENSITIVE. . . . . . .     (Yes/No)
 APPLCOMPAT . . . . . . . . . .     (VnnRn/VnnRnMnnn)
 CONCENTRATE STATEMENTS . . . .     (O-OFF, L-WITH LITERALS)
```

*Figure 154. **Advanced trigger options (ADBP6CJW)** panel*

6. If you specified Yes to create an advanced trigger, specify values for the following parameters:

Use CREATE OR REPLACE
Indicates whether to replace or change the definition for the trigger.

VERSION
The version identifier for the trigger. This field is optional.

DEBUG MODE
Indicates whether this version of the trigger can be run in debug mode. Valid values are D, A, and S.

WLM for DEBUG MODE
The workload manager (WLM) application environment that is used by Db2 when the trigger is debugged. This field is optional.

QUALIFIER
The implicit qualifier that is used for unqualified object names that are referenced in the trigger body. This field is optional.

ASUTIME LIMIT
The total amount of processor time in CPU service units that a single invocation of this version of the trigger can run or 0 to specify that there is no limit on the service units.

CURRENT DATA
Indicates whether data currency is required for read-only and ambiguous cursors.

CONCURRENT ACCESS RESOLUTION
Indicates whether processing uses only committed data or whether it will wait for commit or rollback of data that is in the process of being updated.

DYNAMICRULES
Indicates whether you want the dynamic SQL statements to be processed by using run behavior or bind behavior.

APPLICATION CCSID
The default encoding scheme for SQL variables in static SQL statements in the trigger body. Valid values are A, E, and U.

EXPLAIN
Indicates whether information will be provided about how SQL statements in the trigger will execute.

IMMEDIATE WRITE
Indicates whether immediate writes are to be done for updates that are made to group buffer pool dependent page sets or partitions.

ISOLATION LEVEL
Indicates how far to isolate the trigger from the effects of other running applications. Valid values are C, S, R, and U.

OPTHINT
A character string of up to 128 bytes in length, which is used by the Db2 subsystem when searching the PLAN_TABLE for rows to use as input.

SQL PATH
The SQL path that Db2 uses to resolve unqualified user-defined types, functions, and procedure names in the body of the trigger.

RELEASE AT
Indicates when to release resources that the trigger uses: either at each commit point or when the trigger terminates.

ROUNDING
The rounding mode for manipulation of DECFLOAT data.

DATE FORMAT
The date format for result values that are string representations of date or time values.

TIME FORMAT
The time format for result values that are string representations of date or time values.

DECIMAL PRECISION
The maximum precision that is to be used for decimal arithmetic operations.

SCALE
The minimum scale that is to be used for division.

FOR UPDATE CLAUSE
Indicates whether the FOR UPDATE clause is required for a DECLARE CURSOR statement if the cursor is to be used to perform positioned updates.

BUSINESS TIME SENSITIVE
Indicates whether references to application-period temporal tables in both static and dynamic SQL statements are affected by the value of the CURRENT TEMPORAL BUSINESS_TIME special register.

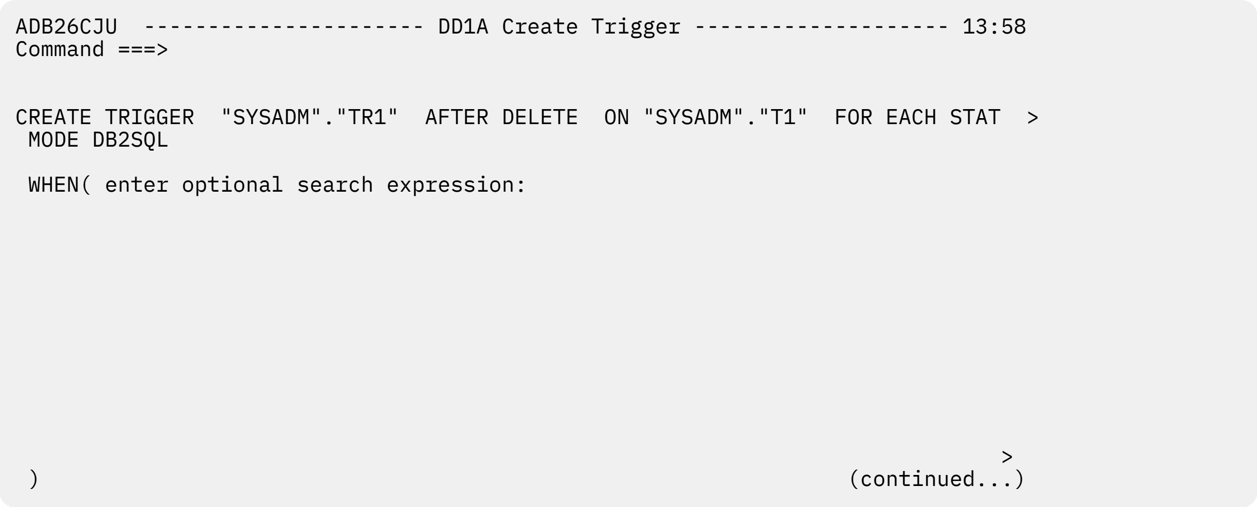
SYSTEM TIME SENSITIVE
Indicates whether references to system-period temporal tables in both static and dynamic SQL statements are affected by the value of the CURRENT TEMPORAL SYSTEM_TIME special register.

ARCHIVE SENSITIVE
Indicates whether references to archive-enabled tables in SQL statements are affected by the value of the SYSIBMADM.GET_ARCHIVE global variable.

APPLCOMPAT
The package compatibility level behavior for static SQL.

CONCENTRATE STATEMENTS
Indicates whether a dynamic SQL statement that specifies literal constants will be cached as a separate unique statement entry in the dynamic statement cache instead of sharing an existing statement in the cache.

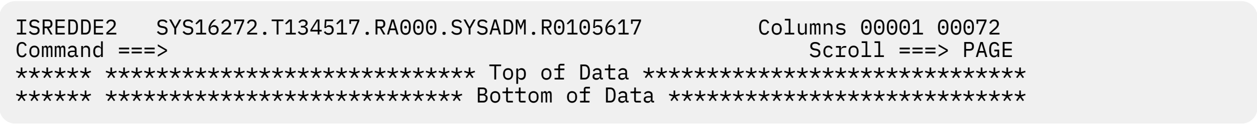
7. Edit the WHEN clause.



```
ADB26CJU  --------------------- DD1A Create Trigger -------------------- 13:58
Command ===>


CREATE TRIGGER  "SYSADM"."TR1"  AFTER DELETE  ON "SYSADM"."T1"  FOR EACH STAT  >
 MODE DB2SQL

 WHEN( enter optional search expression:











                                                                             >
 )                                                               (continued...)
```

Figure 155. Screen to edit the WHEN clause (ADB26CJU)

8. Edit the trigger body.



```
ISREDDE2   SYS16272.T134517.RA000.SYSADM.R0105617         Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
****** ***************************** Top of Data ******************************
****** **************************** Bottom of Data ****************************
```

Figure 156. Screen to edit the trigger body (ISREDDE2)

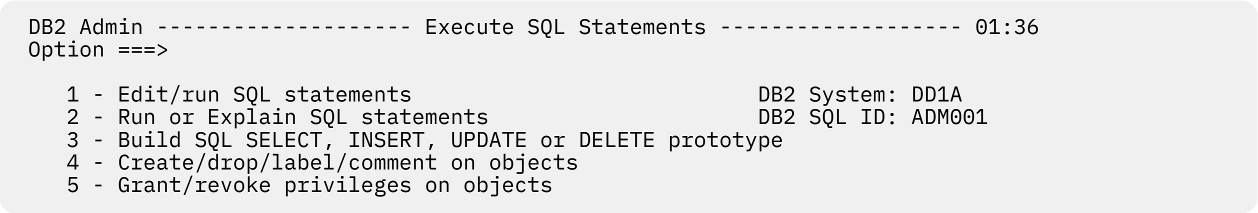
Placing labels on tables

Use the **Label Table (ADB26LT)** panel to place a label on a table.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.

   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure:

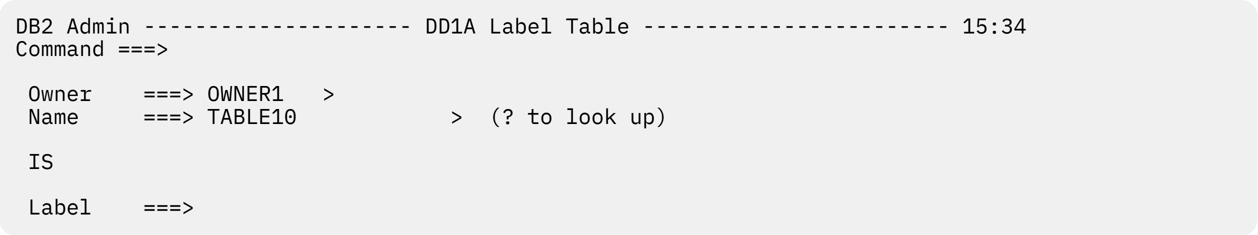


```
DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
Option ===>

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

Figure 157. ***Execute SQL Statements (ADB22)*** *panel*

2. Specify option 4, and press Enter.
3. Specify option LT, and press Enter.

   The **Label Table (ADB26LT)** panel is displayed, as shown in the following figure.



```
DB2 Admin --------------------- DD1A Label Table ------------------------ 15:34
Command ===>

 Owner    ===> OWNER1   >
 Name     ===> TABLE10            >  (? to look up)

 IS

 Label    ===>
```

Figure 158. ***Label Table (ADB26LT)*** *panel*

4. Specify the following values, and press Enter.

   Owner

   An owner name for the table on which you want to place a label.

   Name

   A table name on which you want to place a label, or enter a question mark (?) to look up existing table names using the **Tables, Views, and Aliases (ADB21T)** panel.

   Label

   A label for the table.
5. Follow the instructions on the **Statement Execution Prompt (ADB2PSTM)** panel (if enabled) to complete and run the SQL statement for placing the label on the table.

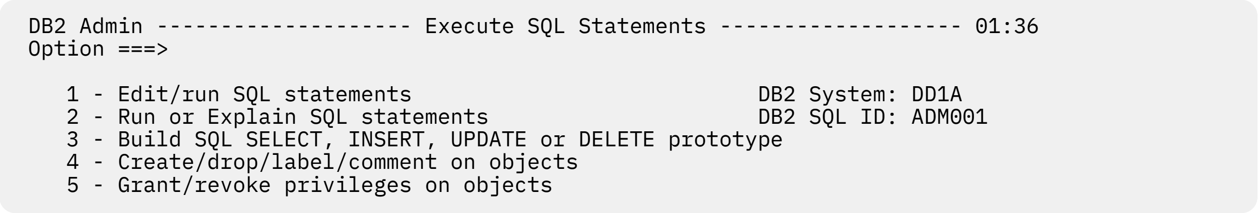
Placing comments on tables

Use the **Comment Table (ADB26RT)** panel to place a comment on a table.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.

   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure:

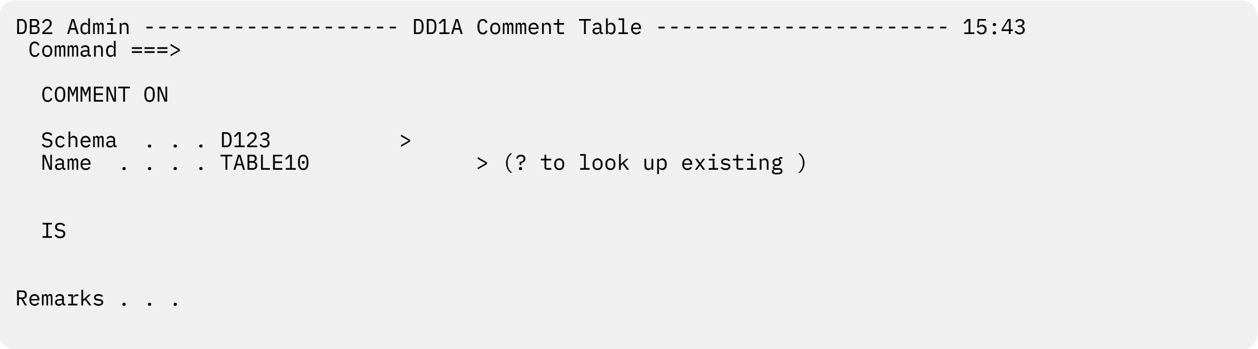


```
DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
Option ===>

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

Figure 159. ***Execute SQL Statements (ADB22)*** *panel*

2. Specify option 4, and press Enter.
3. Specify option RT, and press Enter.

   The **Comment Table (ADB26RT)** panel is displayed, as shown in the following figure.



```
DB2 Admin -------------------- DD1A Comment Table ----------------------- 15:43
 Command ===>

  COMMENT ON

  Schema  . . . D123          >
  Name   . . . . TABLE10            > (? to look up existing )


  IS


Remarks . . .
```

Figure 160. ***Comment Table (ADB26RT)*** *panel*

4. Specify the following values, and press Enter.

   Schema
   A schema that is described in the catalog. This value indicates a comment will be added or replaced for a schema.

   Name
   A table name on which you want to place a comment or enter a question mark (?) to look up existing table names using the **Tables, Views, and Aliases (ADB21T)** panel.

   Remarks
   Your comments about the table.

5. Follow the instructions on the **Statement Execution Prompt (ADB2PSTM)** panel (if enabled) to complete and run the SQL statement for placing the comment on the table.

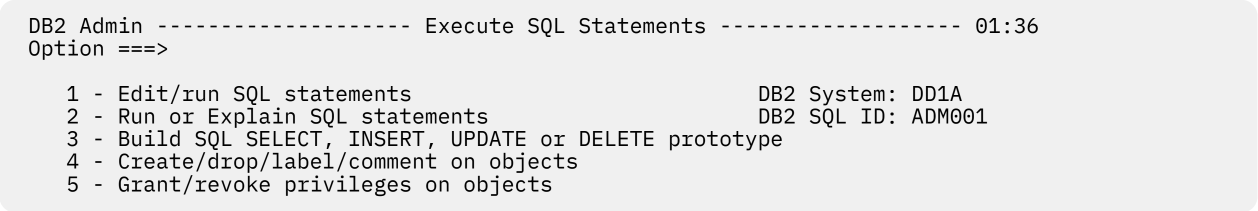
Dropping tables

Use the **Drop Table (ADB26DT)** panel to drop a table.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.

   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure:

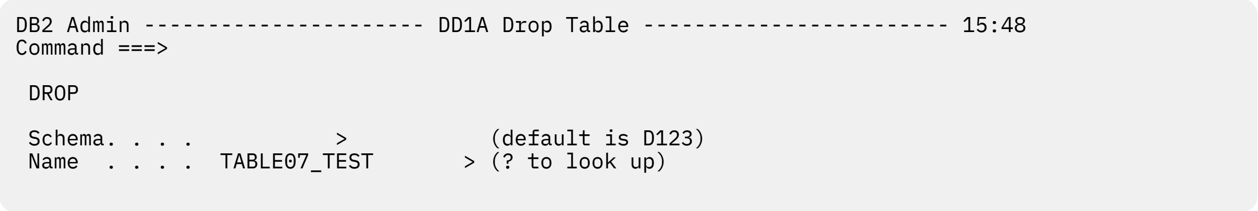


```
 DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
 Option ===>

    1 - Edit/run SQL statements                           DB2 System: DD1A
    2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
    3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
    4 - Create/drop/label/comment on objects
    5 - Grant/revoke privileges on objects
```

*Figure 161. E****Execute SQL Statements (ADB22)*** *panel*

2. Specify option 4, and press Enter.
3. Specify option DT, and press Enter.

   The **Drop Table (ADB26DT)** panel is displayed, as shown in the following figure.



```
DB2 Admin --------------------- DD1A Drop Table ------------------------ 15:48
Command ===>

 DROP

 Schema. . . .           >          (default is D123)
 Name  . . . .  TABLE07_TEST       > (? to look up)
```

Figure 162. ***Drop Table (ADB26DT)*** *panel*

4. Specify the following values, and press Enter.

Schema
: The schema for the table that you want to drop. The name must identify a schema that is described in the catalog.

Name
: A table name that you want to drop or enter a question mark (?) to look up existing table names using the **Tables, Views, and Aliases (ADB21T)** panel.

5. Follow the instructions on the **Statement Execution Prompt (ADB2PSTM)** panel (if enabled) to complete and run the SQL statement for dropping the table.

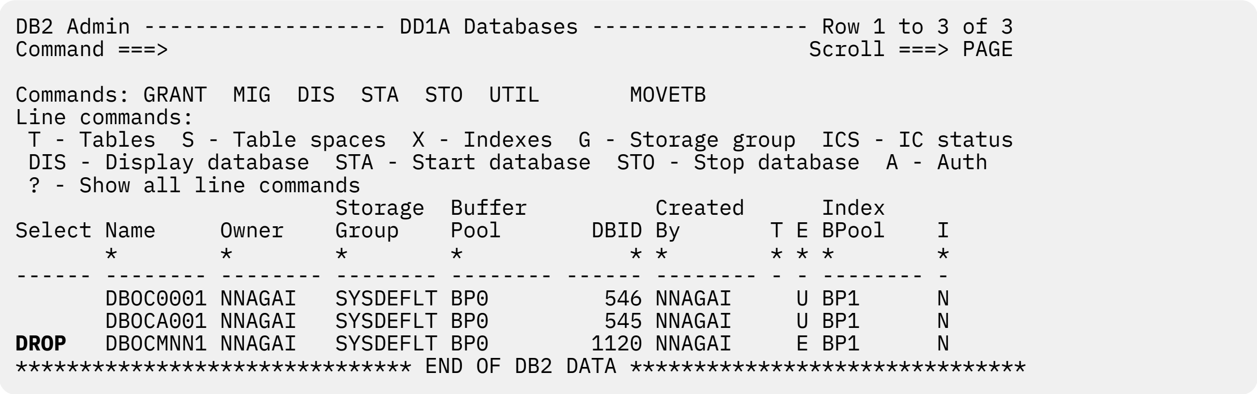
Requesting drop impact reports

Drop impact reports identify other Db2 objects, plans, and packages that are impacted when an object is dropped. These reports are useful tools that can help you avoid dropping objects that adversely impact other objects, plans, and packages. Consider generating a drop impact report whenever you drop an object.

Procedure

To request drop impact reports:

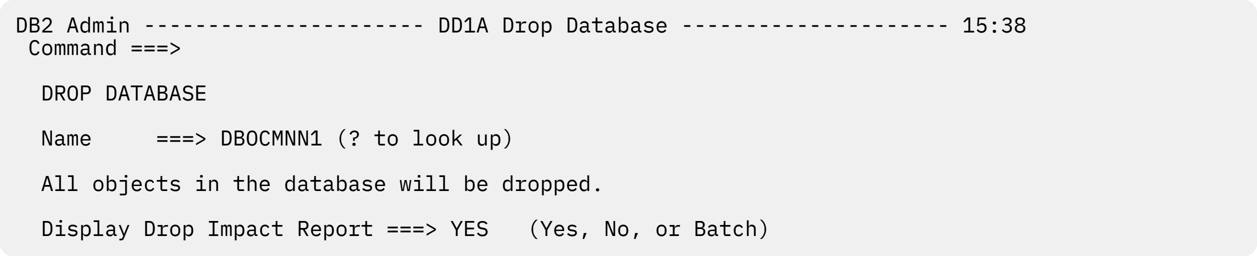
1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify any selection criteria at the bottom, specify option D, and press Enter.
3. On the **Databases (ADB21D)** panel, issue the DROP line command to drop a database and press Enter:



```
DB2 Admin ------------------- DD1A Databases ----------------- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       *        *        *        *             * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
       DBOC0001 NNAGAI   SYSDEFLT BP0         546 NNAGAI     U BP1      N
       DBOCA001 NNAGAI   SYSDEFLT BP0         545 NNAGAI     U BP1      N
DROP   DBOCMNN1 NNAGAI   SYSDEFLT BP0        1120 NNAGAI     E BP1      N
******************************* END OF DB2 DATA *******************************
```

*Figure 163. Using the DROP command on the **Databases (ADB21D)** panel*

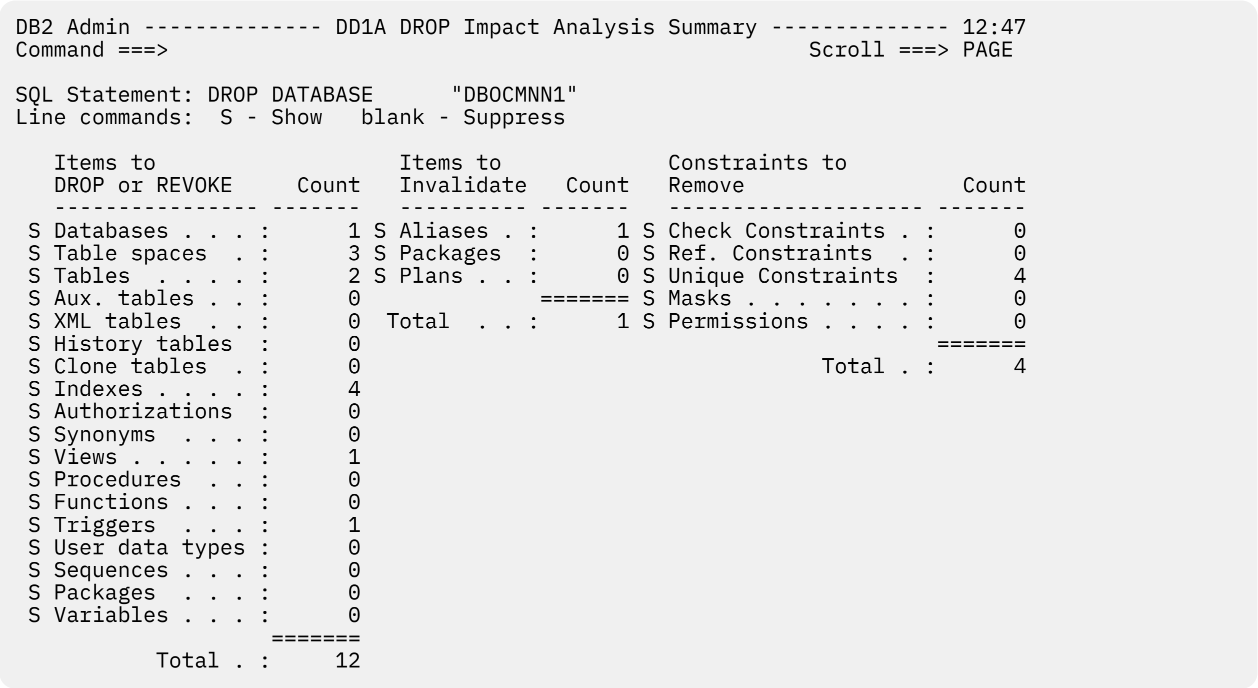
4. On the **Drop Database (ADB26DD)** panel, ensure that the **Display Drop Impact Report** field is set to YES, and press Enter:



```
DB2 Admin --------------------- DD1A Drop Database --------------------- 15:38
 Command ===>

  DROP DATABASE

  Name     ===> DBOCMNN1 (? to look up)

  All objects in the database will be dropped.

  Display Drop Impact Report ===> YES   (Yes, No, or Batch)
```

Figure 164. The Drop Database panel (ADB26DD)

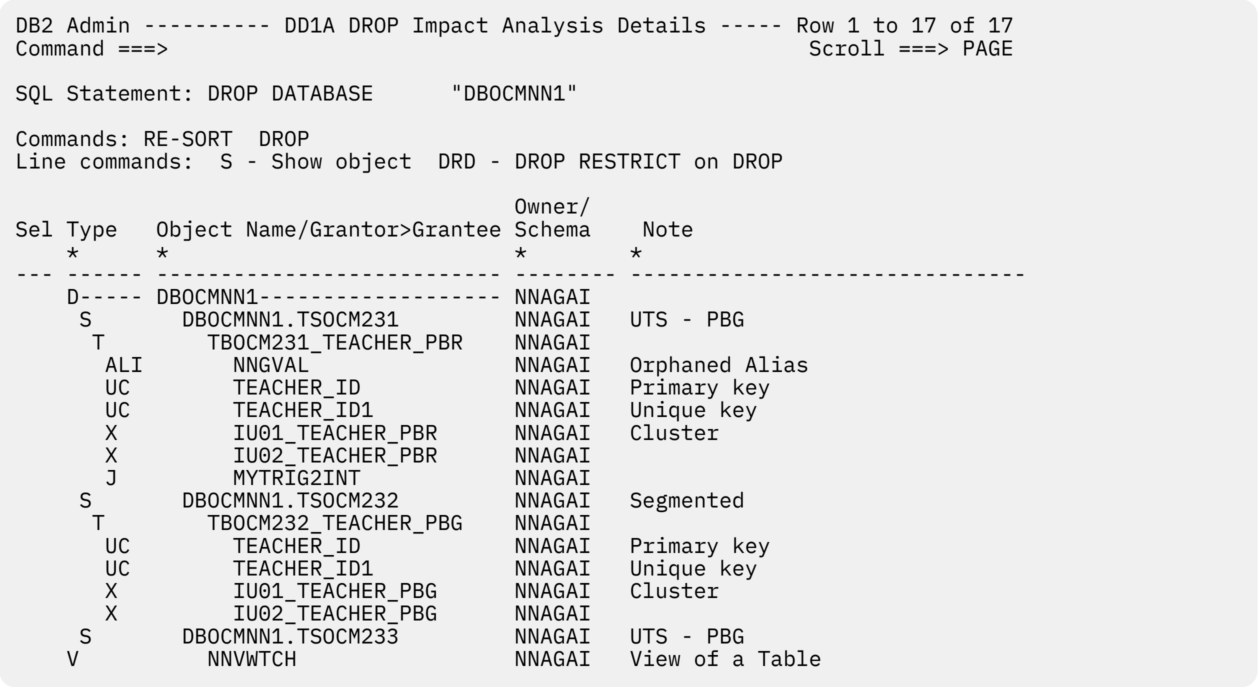
The **DROP Impact Analysis Summary (ADB2DIP)** panel is displayed. The following figure shows a portion of this panel.



```
DB2 Admin -------------- DD1A DROP Impact Analysis Summary -------------- 12:47
Command ===>                                                  Scroll ===> PAGE

SQL Statement: DROP DATABASE      "DBOCMNN1"
Line commands:  S - Show   blank - Suppress

   Items to                   Items to             Constraints to
   DROP or REVOKE     Count   Invalidate   Count   Remove                 Count
   ---------------- -------   ---------- -------   -------------------- -------
 S Databases . . . :      1 S Aliases . :      1 S Check Constraints . :      0
 S Table spaces  . :      3 S Packages  :      0 S Ref. Constraints  . :      0
 S Tables  . . . . :      2 S Plans . . :      0 S Unique Constraints  :      4
 S Aux. tables . . :      0              ======= S Masks . . . . . . . :      0
 S XML tables  . . :      0  Total  . . :      1 S Permissions . . . . :      0
 S History tables  :      0                                             =======
 S Clone tables  . :      0                                    Total . :      4
 S Indexes . . . . :      4
 S Authorizations  :      0
 S Synonyms  . . . :      0
 S Views . . . . . :      1
 S Procedures  . . :      0
 S Functions . . . :      0
 S Triggers  . . . :      1
 S User data types :      0
 S Sequences . . . :      0
 S Packages  . . . :      0
 S Variables . . . :      0
                    =======
        Total . :     12
```

*Figure 165. **DROP Impact Analysis Summary (ADB2DIP)** panel*

5. Press Enter to display the **DROP Impact Analysis Details (ADB2DIPD)** panel. This panel displays all objects that are impacted by dropping the object. The following figure shows a portion of this panel.



```
DB2 Admin ---------- DD1A DROP Impact Analysis Details ----- Row 1 to 17 of 17
Command ===>                                                  Scroll ===> PAGE

SQL Statement: DROP DATABASE      "DBOCMNN1"

Commands: RE-SORT  DROP
Line commands:  S - Show object  DRD - DROP RESTRICT on DROP

                                       Owner/
Sel Type   Object Name/Grantor>Grantee Schema    Note
    *      *                           *        *
--- ------ --------------------------- -------- -------------------------------
    D----- DBOCMNN1------------------- NNAGAI
     S       DBOCMNN1.TSOCM231         NNAGAI   UTS - PBG
      T        TBOCM231_TEACHER_PBR    NNAGAI
       ALI       NNGVAL                NNAGAI   Orphaned Alias
       UC        TEACHER_ID            NNAGAI   Primary key
       UC        TEACHER_ID1           NNAGAI   Unique key
       X         IU01_TEACHER_PBR      NNAGAI   Cluster
       X         IU02_TEACHER_PBR      NNAGAI
       J         MYTRIG2INT            NNAGAI
     S       DBOCMNN1.TSOCM232         NNAGAI   Segmented
      T        TBOCM232_TEACHER_PBG    NNAGAI
       UC        TEACHER_ID            NNAGAI   Primary key
       UC        TEACHER_ID1           NNAGAI   Unique key
       X         IU01_TEACHER_PBG      NNAGAI   Cluster
       X         IU02_TEACHER_PBG      NNAGAI
     S       DBOCMNN1.TSOCM233         NNAGAI   UTS - PBG
    V          NNVWTCH                 NNAGAI   View of a Table
```

*Figure 166. Partial display of the **DROP Impact Analysis Details (ADB2DIPD)** panel*

On the **DROP Impact Analysis Details (ADB2DIPD)** panel, you can issue the following commands:

Primary commands

RE-SORT
Re-sorts the table to its original sequence.

DROP
Proceed to drop the object.

Restriction: On the panel, you must type the Drop command on the primary command line and press Enter. You cannot issue the command by positioning the cursor on the DROP primary command and pressing Enter.

Sort
Sort the table based on one or more columns.

Line commands

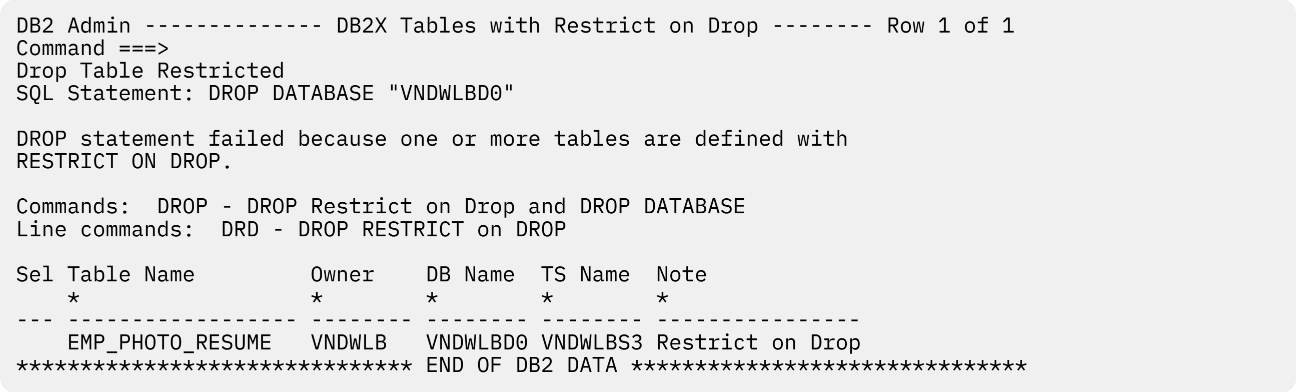
S
Shows more details about an object.

DRD
Drop Restrict on Drop for the object.

The Restrict on Drop attribute

If a table has the Restrict on Drop attribute, users are restricted from dropping the object until the attribute is removed.

Occasionally, Db2 tables contain the Restrict on Drop attribute to prevent users from accidentally dropping them. When attempting to drop one or more tables that have the Restrict on Drop attribute, Db2 Admin Tool displays the **Tables with Restrict on Drop** panel, as shown in the following figure.



```
DB2 Admin -------------- DB2X Tables with Restrict on Drop -------- Row 1 of 1
Command ===>
Drop Table Restricted
SQL Statement: DROP DATABASE "VNDWLBD0"

DROP statement failed because one or more tables are defined with
RESTRICT ON DROP.

Commands:  DROP - DROP Restrict on Drop and DROP DATABASE
Line commands:  DRD - DROP RESTRICT on DROP

Sel Table Name         Owner    DB Name  TS Name  Note
    *                  *        *        *        *
--- ------------------ -------- -------- -------- ----------------
    EMP_PHOTO_RESUME   VNDWLB   VNDWLBD0 VNDWLBS3 Restrict on Drop
******************************* END OF DB2 DATA *******************************
```

*Figure 167. **Tables with Restrict on Drop** panel (ADB26DDR)*

Db2 Admin Tool also displays this panel if a user attempts to drop a database or a table space that contains one or more tables that have the Restrict on Drop attribute.

To remove the Restrict on Drop attribute from a table, use the DRD line command. The DRD line command removes the Restrict on Drop attribute without dropping the table.

When dropping a database, table space, or table, you can use the DROP primary command to remove the Restrict on Drop attribute from the tables that are involved and then drop the database, table space, or table.

When the PROMPT option is used while dropping an object, the DROP statement for the object is displayed. You must select option 1 to run the DROP statement. If the DROP statement fails (with error code −672) because one or more tables have the Restrict on Drop attribute, the **Tables with Restrict on Drop** (ADB26DDR) panel is displayed. At this point, you can take one of the following actions:

- Use the DROP primary command to remove the Restrict on Drop attribute from the tables and run the DROP statement again. The DROP primary command generates an ALTER DROP RESTRICT ON DROP statement for each table, followed by a DROP statement.
- Use the DRD line command to remove the Restrict on Drop attribute for an individual table.
- Cancel and exit without running the DROP statement.

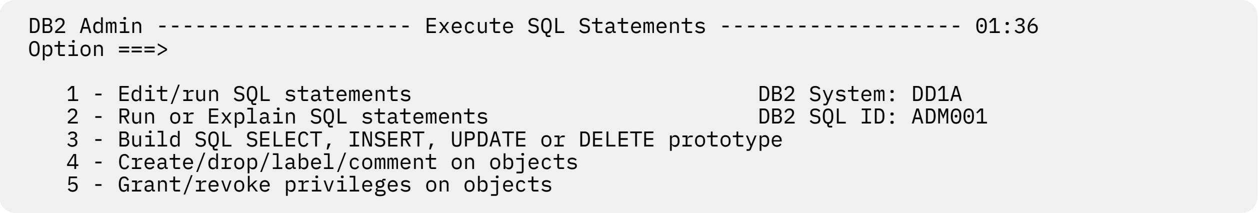
Creating a global variable

Use the **Create Global Variable (ADBP6CGV)** panel to create a new global variable.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.

   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure:

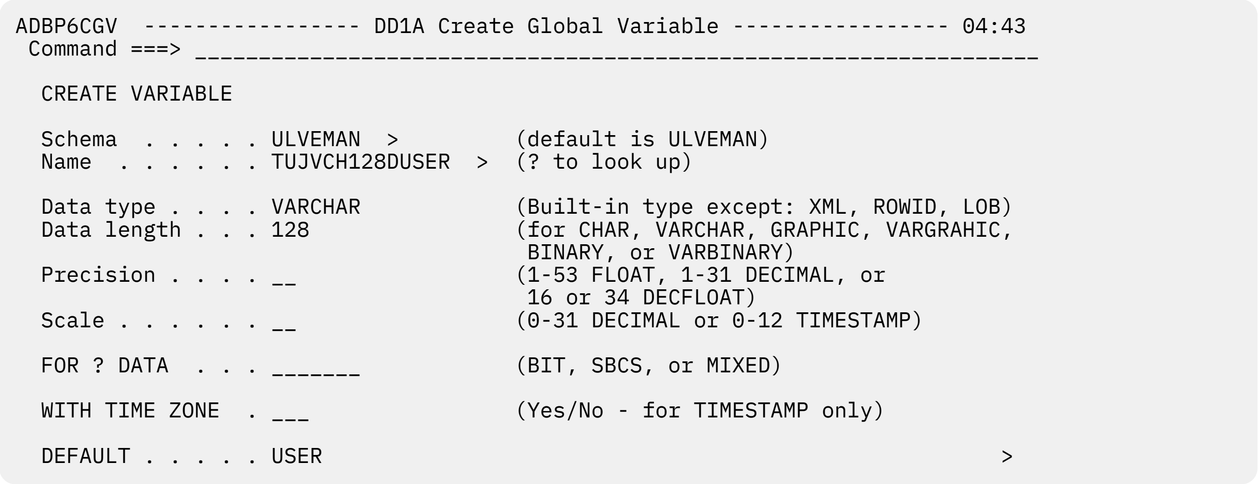


```
 DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
 Option ===>

    1 - Edit/run SQL statements                           DB2 System: DD1A
    2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
    3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
    4 - Create/drop/label/comment on objects
    5 - Grant/revoke privileges on objects
```

   Figure 168. Execute SQL Statements panel (ADB22)

2. Specify option 4, and press Enter.
3. Specify option CGV, and press Enter.

   The **Create Global Variable (ADBP6CGV)** panel is displayed, as shown in the following figure.



```
ADBP6CGV  ----------------- DD1A Create Global Variable ----------------- 04:43
 Command ===> __________________________________________________________________

  CREATE VARIABLE

  Schema  . . . . . ULVEMAN  >         (default is ULVEMAN)
  Name  . . . . . . TUJVCH128DUSER  >  (? to look up)

  Data type . . . . VARCHAR            (Built-in type except: XML, ROWID, LOB)
  Data length . . . 128                (for CHAR, VARCHAR, GRAPHIC, VARGRAHIC,
                                        BINARY, or VARBINARY)
  Precision . . . . __                 (1-53 FLOAT, 1-31 DECIMAL, or
                                        16 or 34 DECFLOAT)
  Scale . . . . . . __                 (0-31 DECIMAL or 0-12 TIMESTAMP)

  FOR ? DATA  . . . _______            (BIT, SBCS, or MIXED)

  WITH TIME ZONE  . ___                (Yes/No - for TIMESTAMP only)

  DEFAULT . . . . . USER                                                      >
```

   *Figure 169. **Create Global Variable (ADBP6CGV)** panel*

4. Specify the following values for the global variable, and press Enter.

   Schema
   The schema.

   Name
   The name.

   Data type
   The data type.

   Restriction: XML, ROWID, or LOB data types are not valid in this field.

   Data length
   The maximum length.

   Precision
   The precision. Precision only applies to FLOAT, DECIMAL, or DECFLOAT data types.

   Scale
   The scale. Scale only applies to DECIMAL, or TIMESTAMP data types.

   FOR ? DATA
   If applicable, the subtype for a CHARACTER data type.

WITH TIME ZONE
If applicable, the subtype for a TIMESTAMP data type.

Default
The default value.

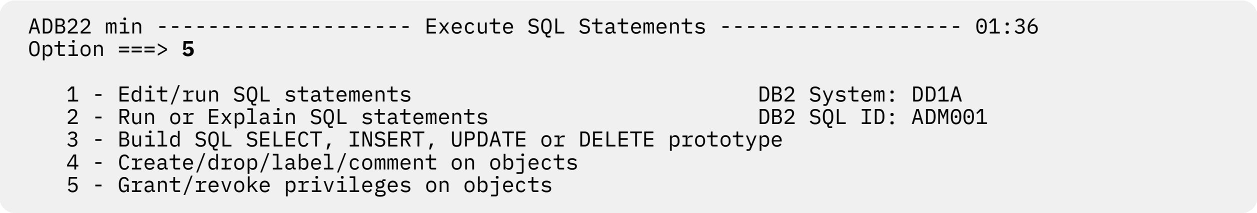
5. Follow the instructions on the Statement Execution Prompt panel (if enabled) to complete and run the SQL statement for creating the new global variable.

Revoking privileges

Db2 Admin Tool guides you through the process of revoking privileges without requiring you to know the syntax of the REVOKE SQL statement.

Procedure

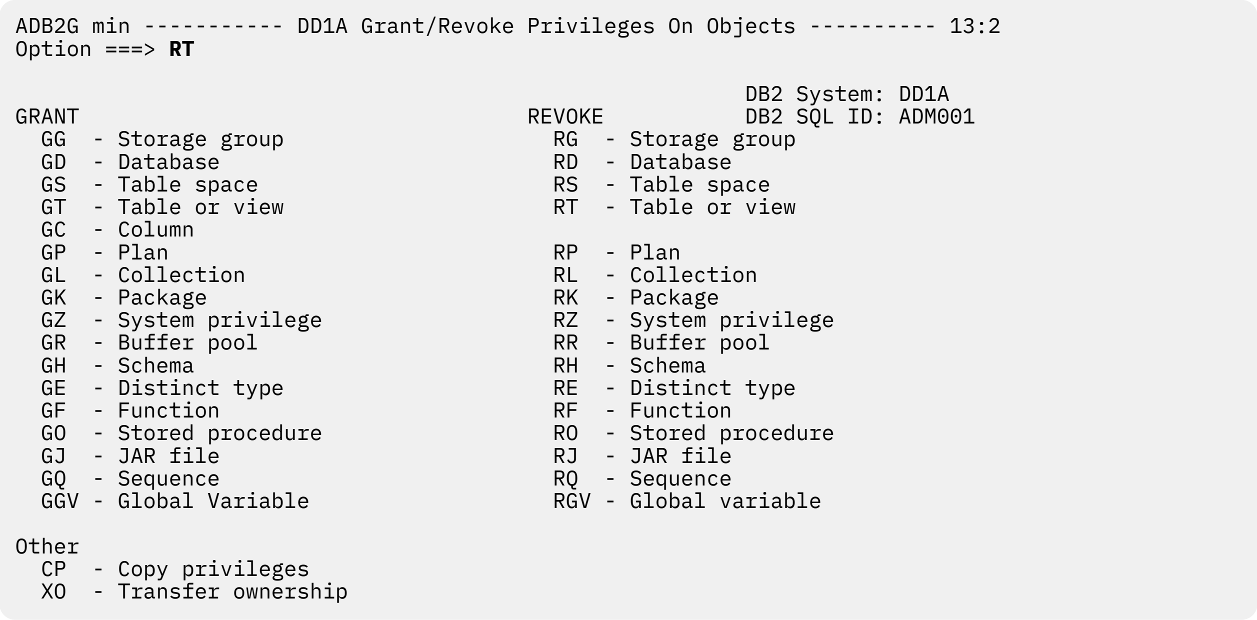
1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.
2. On the **Execute SQL Statements (ADB22)** panel, specify option 5 and press Enter.



```
ADB22 min -------------------- Execute SQL Statements ------------------- 01:36
Option ===> 5

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

*Figure 170. **Execute SQL Statements (ADB22)** panel*

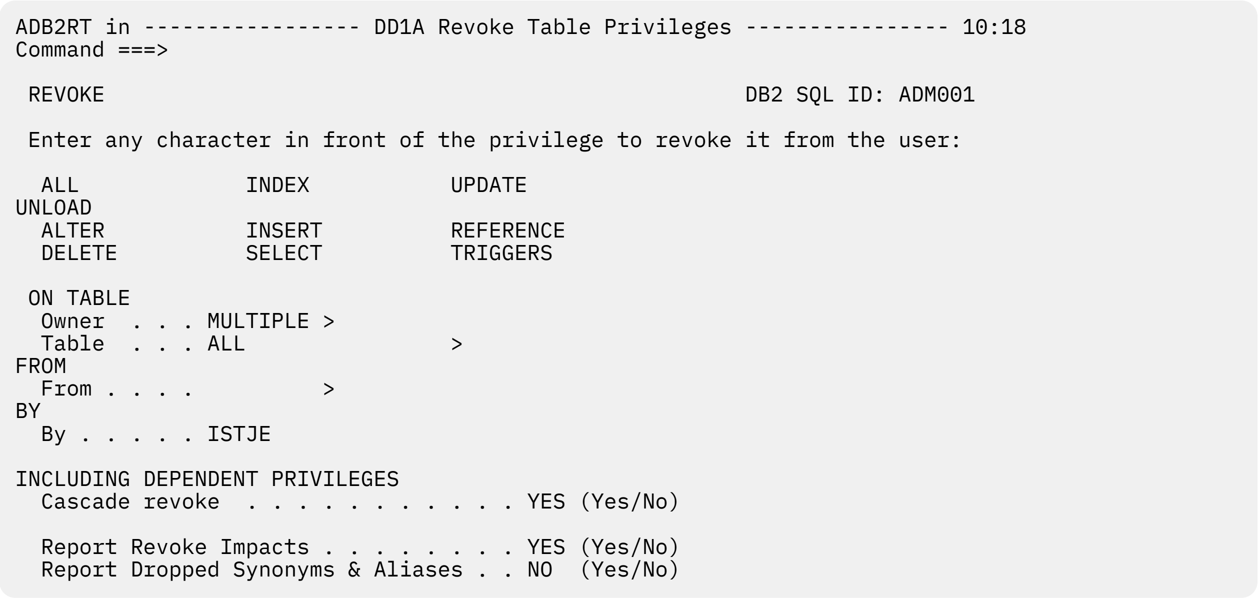
3. On the **Grant/Revoke Privileges On Objects (ADB2G)** panel, specify RT, and press Enter:



```
ADB2G min ----------- DD1A Grant/Revoke Privileges On Objects ---------- 13:2
Option ===> RT

                                                         DB2 System: DD1A
GRANT                                  REVOKE           DB2 SQL ID: ADM001
  GG  - Storage group                    RG  - Storage group
  GD  - Database                         RD  - Database
  GS  - Table space                      RS  - Table space
  GT  - Table or view                    RT  - Table or view
  GC  - Column
  GP  - Plan                             RP  - Plan
  GL  - Collection                       RL  - Collection
  GK  - Package                          RK  - Package
  GZ  - System privilege                 RZ  - System privilege
  GR  - Buffer pool                      RR  - Buffer pool
  GH  - Schema                           RH  - Schema
  GE  - Distinct type                    RE  - Distinct type
  GF  - Function                         RF  - Function
  GO  - Stored procedure                 RO  - Stored procedure
  GJ  - JAR file                         RJ  - JAR file
  GQ  - Sequence                         RQ  - Sequence
  GGV - Global Variable                  RGV - Global variable

Other
  CP  - Copy privileges
  XO  - Transfer ownership
```

*Figure 171. **Grant/Revoke Privileges On Objects (ADB2G)** panel*

4. On the **Revoke Table Privileges (ADB2RT)** panel, complete the fields and press Enter to revoke the selected privilege:



```
ADB2RT in ----------------- DD1A Revoke Table Privileges ---------------- 10:18
Command ===>

 REVOKE                                                  DB2 SQL ID: ADM001

 Enter any character in front of the privilege to revoke it from the user:

  ALL             INDEX           UPDATE
UNLOAD
  ALTER           INSERT          REFERENCE
  DELETE          SELECT          TRIGGERS

 ON TABLE
  Owner  . . . MULTIPLE >
  Table  . . . ALL                >
FROM
  From . . . .          >
BY
  By . . . . . ISTJE

INCLUDING DEPENDENT PRIVILEGES
  Cascade revoke  . . . . . . . . . . . YES (Yes/No)

  Report Revoke Impacts . . . . . . . . YES (Yes/No)
  Report Dropped Synonyms & Aliases . . NO  (Yes/No)
```

Note: The UNLOAD privilege is displayed only if you are running Db2 12 for z/OS.

Figure 172. ***Revoke Table Privileges (ADB2RT)*** *panel*

- In the list of privileges, select the one that you want to revoke.
- In the **ON TABLE** section, specify the qualified name of the table.
- In the **FROM** field, specify the user ID from which you want to revoke the privilege.
- In the **BY** field, specify the user ID that is revoking the privilege.
- In the **Report Revoke Impacts** field, specify whether you want a report that shows the impact of the REVOKE statement. This report is displayed as a tree structure, which represents all of the authorizations or objects that will be lost or invalidated as a result of the REVOKE statement.
- In the **Report Dropped Synonyms & Aliases** field, specify whether you want a report that shows the synonyms and aliases that will be dropped as a result of the REVOKE statement.

Related tasks

"Requesting revoke impact reports in batch" on page 517
A *revoke impact report* helps you determine how the authorizations and database objects will be affected by revoking an authorization before you actually revoke it. You can request these reports on the **Revoke *object* Privileges** panels. You can also request these reports by using a batch job.

Copying privileges from existing objects to other objects

Use the **Copy privileges (ADBPCP)** panel to copy privileges from existing objects to other objects.

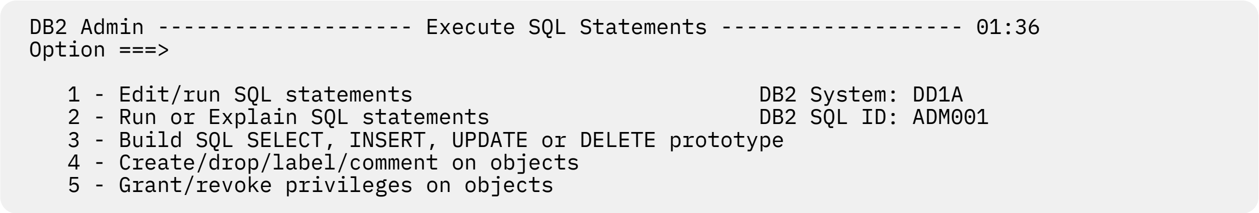
About this task

When new objects are created, you often need to grant privileges to the new objects, and often the same privileges from an existing object are needed for the new objects. The following example shows how to copy privileges from existing objects to other objects.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.

   The **Execute SQL Statements (ADB22)** panel is displayed, as shown in the following figure:

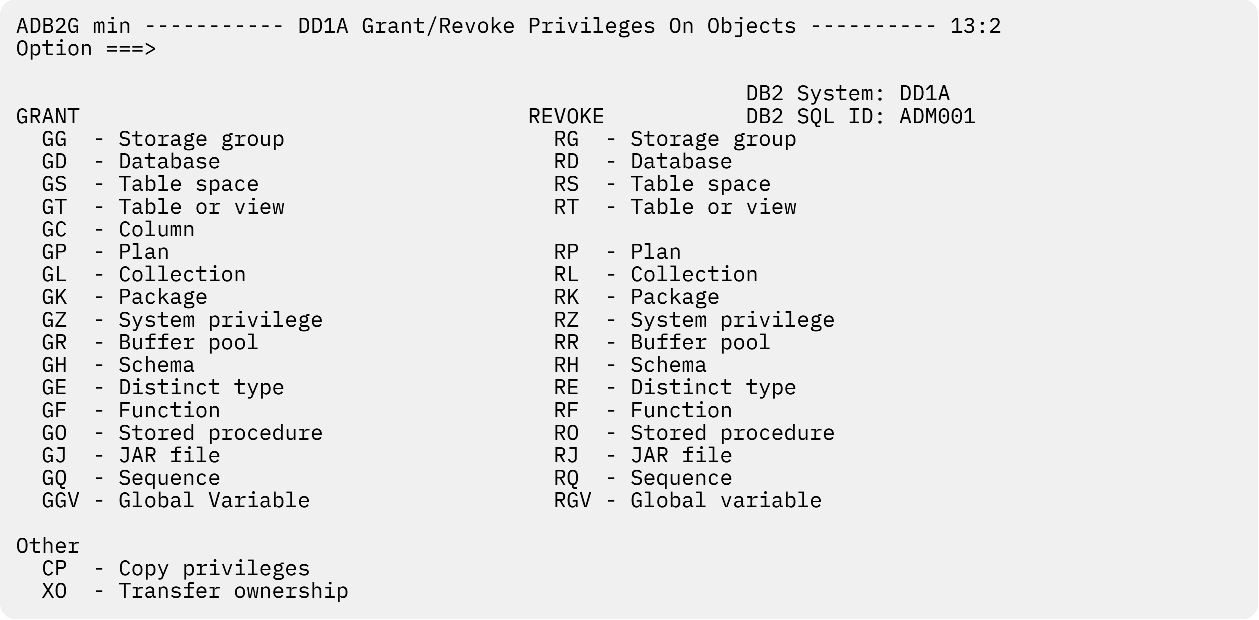


```
DB2 Admin -------------------- Execute SQL Statements ------------------- 01:36
Option ===>

   1 - Edit/run SQL statements                           DB2 System: DD1A
   2 - Run or Explain SQL statements                     DB2 SQL ID: ADM001
   3 - Build SQL SELECT, INSERT, UPDATE or DELETE prototype
   4 - Create/drop/label/comment on objects
   5 - Grant/revoke privileges on objects
```

Figure 173. ***Execute SQL Statements (ADB22)*** *panel*

2. Specify option 5, and press Enter.

   The **Grant/Revoke Privileges On Objects (ADB2G)** panel is displayed, as shown in the following figure.



```
ADB2G min ----------- DD1A Grant/Revoke Privileges On Objects ---------- 13:2
Option ===>

                                                         DB2 System: DD1A
GRANT                                   REVOKE           DB2 SQL ID: ADM001
  GG  - Storage group                     RG  - Storage group
  GD  - Database                          RD  - Database
  GS  - Table space                       RS  - Table space
  GT  - Table or view                     RT  - Table or view
  GC  - Column
  GP  - Plan                              RP  - Plan
  GL  - Collection                        RL  - Collection
  GK  - Package                           RK  - Package
  GZ  - System privilege                  RZ  - System privilege
  GR  - Buffer pool                       RR  - Buffer pool
  GH  - Schema                            RH  - Schema
  GE  - Distinct type                     RE  - Distinct type
  GF  - Function                          RF  - Function
  GO  - Stored procedure                  RO  - Stored procedure
  GJ  - JAR file                          RJ  - JAR file
  GQ  - Sequence                          RQ  - Sequence
  GGV - Global Variable                   RGV - Global variable

Other
  CP  - Copy privileges
  XO  - Transfer ownership
```

Figure 174. ***Grant/Revoke Privileges On Objects (ADB2G)*** *panel*

3. Specify option CP on the associated panel to copy privileges from the following object types, and press Enter.

   Aliases (ADB21A) panel
   Storage Groups (ADB21G) panel
   Databases (ADB21D) panel
   Table Spaces (ADB21S) panel
   Tables, Views, and Aliases (ADB21T) panel
   Schemas (ADB21H) panel
   Data Types (ADB21E) panel
   Functions (ADB21F) panel
   Stored Procedures (ADB21O) panel
   Sequence Objects (ADB21Q) panel
   Grant/Revoke Privileges On Objects (ADB2G) panel
   Version Scopes (ADB2C42) panel
   Global Variables (ADBP1GV) panel

   The **Copy privileges (ADBPCP)** panel.

4. Choose one of the following methods of copying privileges.

   One-to-one
   All privileges from one source object are granted to one target object of the same kind. One-to-one results in GRANT statements for all privileges on one specific object to be built for granting authority to another specific object.

One-to-many

All privileges from one source object are granted to multiple target objects of the same kind. One-to-many results in GRANT statements for all privileges on one specific object to be repeated for each of many other specific objects.

Many-to-many

All privileges for each object in one set of source objects are granted to their counterpart objects in one set of target objects. Many-to-many results in GRANT statements for all privileges on a set of objects and their descendent objects to be built for granting authority to another set of objects.

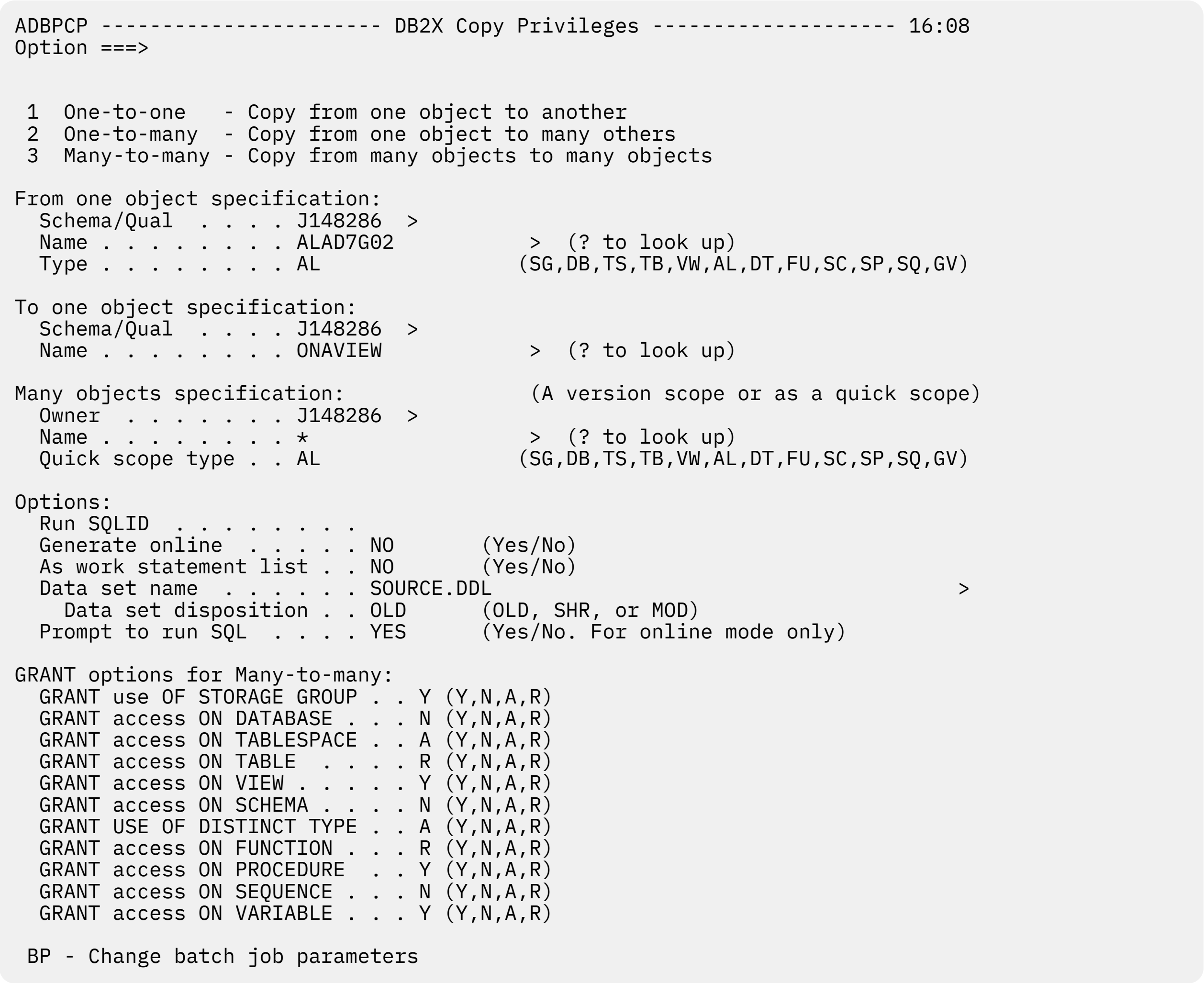
Note: The many-to-many method can produce GRANT statements to non-existent objects. When performed, these GRANTs produce SQLCODE -204, which is tolerated (because of the --#SET ACCEPT_RC statement which precedes these GRANT statements) and processing continues. You can leave these GRANTs in the DDL file (along with the --#SET statements) or remove them.

Restriction: If copy privileges are copied from source objects, for example, OBJECT1 to OBJECT2, OBJECT1 can have a maximum of 30000 GRANTS on it. More than 30000 GRANTS will not be processed as the stack allocated is 30000.

This table shows more detail about the three copying privileges methods:

Table 21. Three methods for copying privileges

	To One	**To Many**
From One	GRANTs from a single object are produced. Source object is provided on the panel. Target object is provided on the panel. No cascading the operation to dependent objects occurs.	GRANTs from a single object are produced. Source object is provided on the panel. Target objects are located by a version scope or quick-version scope. No cascading the operation to dependent objects occurs.
From Many		GRANTs from multiple objects and their dependent objects are produced. Source objects are located by a version scope or quick-version scope. Target objects are determined by masking the source object names. GRANTs to certain object types can be excluded.



```
ADBPCP ----------------------- DB2X Copy Privileges -------------------- 16:08
Option ===>


 1  One-to-one   - Copy from one object to another
 2  One-to-many  - Copy from one object to many others
 3  Many-to-many - Copy from many objects to many objects

From one object specification:
  Schema/Qual  . . . . J148286  >
  Name . . . . . . . . ALAD7G02           >  (? to look up)
  Type . . . . . . . . AL                (SG,DB,TS,TB,VW,AL,DT,FU,SC,SP,SQ,GV)

To one object specification:
  Schema/Qual  . . . . J148286  >
  Name . . . . . . . . ONAVIEW            >  (? to look up)

Many objects specification:               (A version scope or as a quick scope)
  Owner  . . . . . . . J148286  >
  Name . . . . . . . . *                  >  (? to look up)
  Quick scope type . . AL                (SG,DB,TS,TB,VW,AL,DT,FU,SC,SP,SQ,GV)

Options:
  Run SQLID  . . . . . . . .
  Generate online  . . . . . NO       (Yes/No)
  As work statement list . . NO       (Yes/No)
  Data set name  . . . . . . SOURCE.DDL                                      >
    Data set disposition . . OLD      (OLD, SHR, or MOD)
  Prompt to run SQL  . . . . YES      (Yes/No. For online mode only)

GRANT options for Many-to-many:
  GRANT use OF STORAGE GROUP . . Y (Y,N,A,R)
  GRANT access ON DATABASE . . . N (Y,N,A,R)
  GRANT access ON TABLESPACE . . A (Y,N,A,R)
  GRANT access ON TABLE  . . . . R (Y,N,A,R)
  GRANT access ON VIEW . . . . . Y (Y,N,A,R)
  GRANT access ON SCHEMA . . . . N (Y,N,A,R)
  GRANT USE OF DISTINCT TYPE . . A (Y,N,A,R)
  GRANT access ON FUNCTION . . . R (Y,N,A,R)
  GRANT access ON PROCEDURE  . . Y (Y,N,A,R)
  GRANT access ON SEQUENCE . . . N (Y,N,A,R)
  GRANT access ON VARIABLE . . . Y (Y,N,A,R)

 BP - Change batch job parameters
```

Figure 175. ***Copy privileges (ADBPCP)*** *panel*

Revoking system authority from an SQLID

Before you revoke system authority from an SQLID, you can run a Revoke Impact Report.

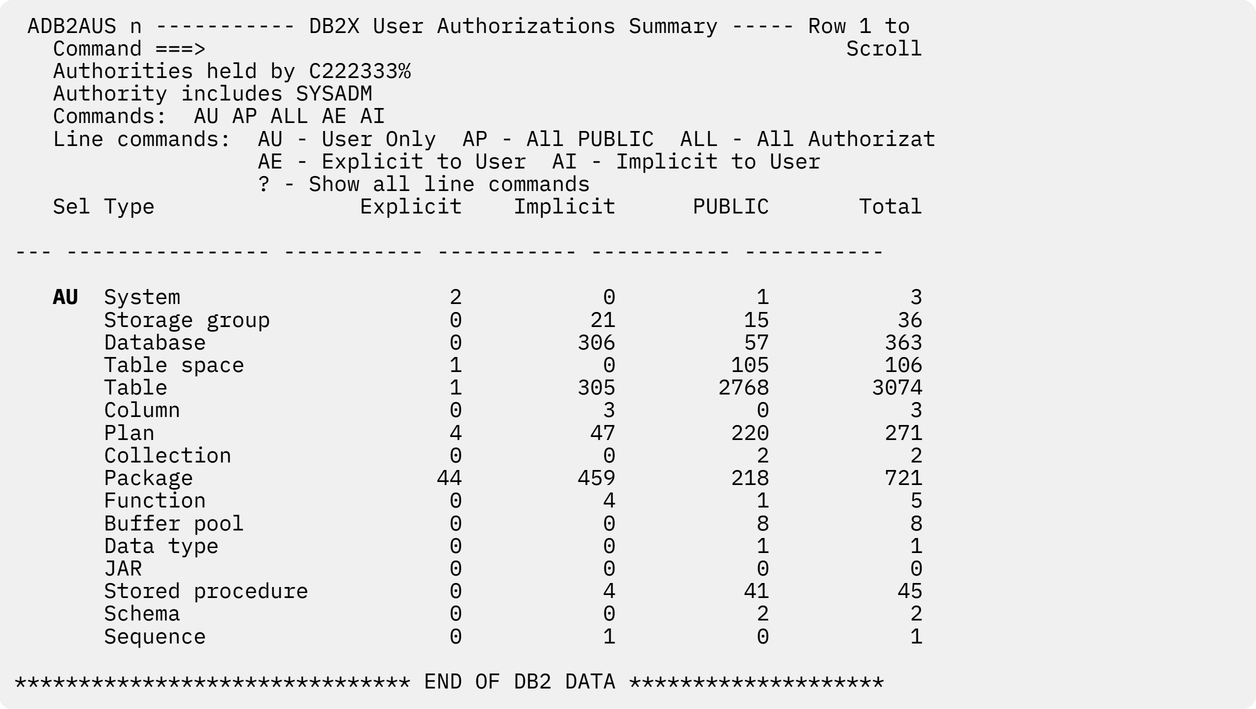
Procedure

To revoke system authority from an SQLID:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, in the **Grantee** field, specify the authorization ID with which you want to revoke, specify the AO command, and press Enter.

   The authorization options are displayed on the **System Catalog (ADB21)** panel.
3. Specify the UA command, and press Enter.

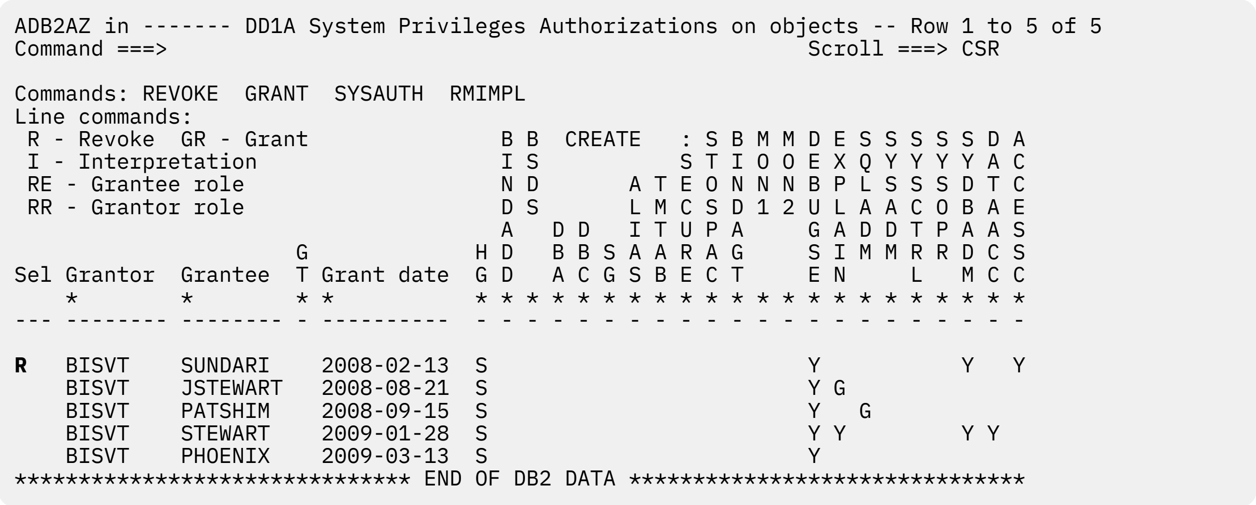
   A summary for the SQL ID is displayed on the **User Authorizations Summary (ADB2AUS)** panel.
4. Specify the AU line command, and press Enter:



```
 ADB2AUS n ----------- DB2X User Authorizations Summary ----- Row 1 to
   Command ===>                                                  Scroll
   Authorities held by C222333%
   Authority includes SYSADM
   Commands:  AU AP ALL AE AI
   Line commands:  AU - User Only  AP - All PUBLIC  ALL - All Authorizat
                   AE - Explicit to User  AI - Implicit to User
                   ? - Show all line commands
   Sel Type                Explicit    Implicit      PUBLIC       Total

--- ---------------- ----------- ----------- ----------- -----------

   AU  System                      2           0           1           3
       Storage group               0          21          15          36
       Database                    0         306          57         363
       Table space                 1           0         105         106
       Table                       1         305        2768        3074
       Column                      0           3           0           3
       Plan                        4          47         220         271
       Collection                  0           0           2           2
       Package                    44         459         218         721
       Function                    0           4           1           5
       Buffer pool                 0           0           8           8
       Data type                   0           0           1           1
       JAR                         0           0           0           0
       Stored procedure            0           4          41          45
       Schema                      0           0           2           2
       Sequence                    0           1           0           1

******************************* END OF DB2 DATA ********************
```

Figure 176. ***User Authorizations Summary (ADB2AUS)*** *panel*

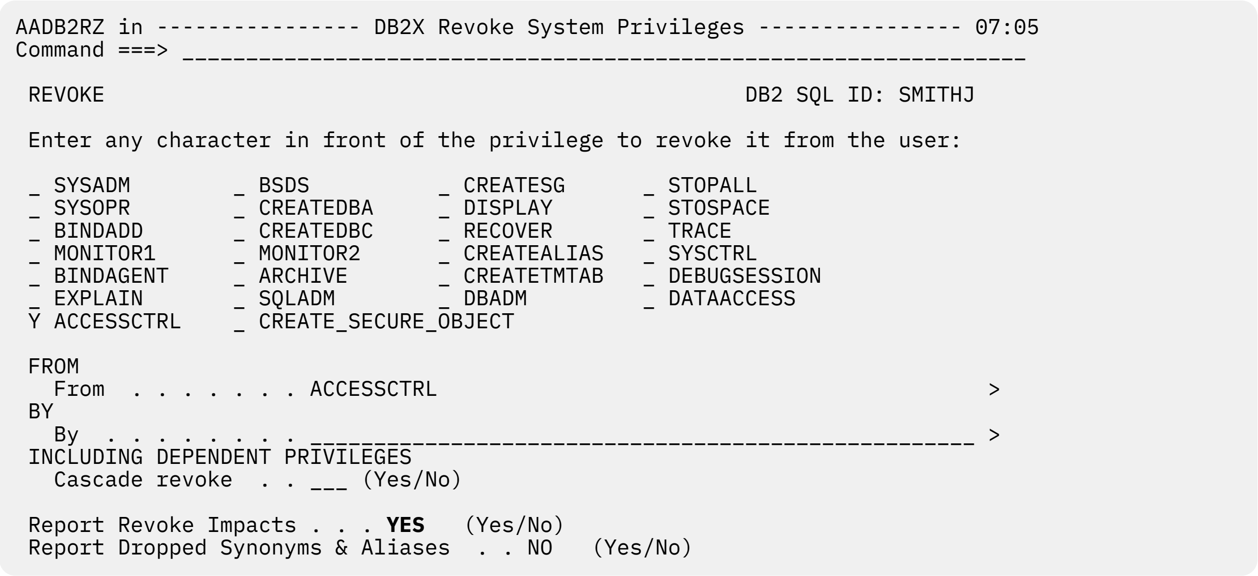
5. On the **System Privileges Authorizations (ADB2AZ)** panel, specify the R line command and press Enter. This actions starts the revoke process and its associated Revoke Impact Report.



```
ADB2AZ in ------- DD1A System Privileges Authorizations on objects -- Row 1 to 5 of 5
Command ===>                                                  Scroll ===> CSR

Commands: REVOKE  GRANT  SYSAUTH  RMIMPL
Line commands:
 R - Revoke  GR - Grant               B B  CREATE   : S B M M D E S S S S S D A
 I - Interpretation                   I S           S T I O O E X Q Y Y Y Y A C
 RE - Grantee role                    N D       A T E O N N N B P L S S S D T C
 RR - Grantor role                    D S       L M C S D 1 2 U L A A C O B A E
                                      A   D D   I T U P A     G A D D T P A A S
                      G             H D   B B S A A R A G     S I M M R R D C S
Sel Grantor  Grantee  T Grant date  G D   A C G S B E C T     E N     L   M C C
    *        *        * *           * * * * * * * * * * * * * * * * * * * * * *
--- -------- -------- - ----------  - - - - - - - - - - - - - - - - - - - - - -

R   BISVT    SUNDARI    2008-02-13  S                         Y           Y   Y
    BISVT    JSTEWART   2008-08-21  S                         Y G
    BISVT    PATSHIM    2008-09-15  S                         Y   G
    BISVT    STEWART    2009-01-28  S                         Y Y         Y Y
    BISVT    PHOENIX    2009-03-13  S                         Y
******************************* END OF DB2 DATA *******************************
```

Figure 177. ***System Privileges Authorizations (ADB2AZ)*** *panel*

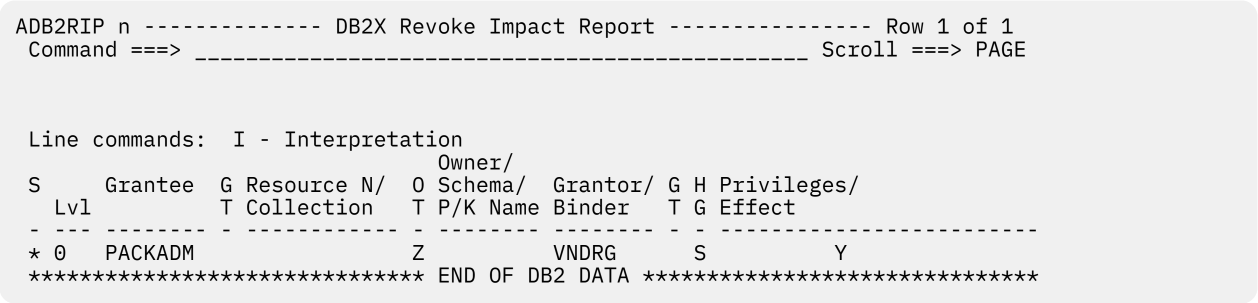
6. On the **Revoke System Privileges (ADB2RZ)** panel, in the **Report Revoke Impacts** field, specify YES, and press Enter.



```
AADB2RZ in ---------------- DB2X Revoke System Privileges ---------------- 07:05
Command ===> ___________________________________________________________________

 REVOKE                                                  DB2 SQL ID: SMITHJ

 Enter any character in front of the privilege to revoke it from the user:

 _ SYSADM        _ BSDS          _ CREATESG      _ STOPALL
 _ SYSOPR        _ CREATEDBA     _ DISPLAY       _ STOSPACE
 _ BINDADD       _ CREATEDBC     _ RECOVER       _ TRACE
 _ MONITOR1      _ MONITOR2      _ CREATEALIAS   _ SYSCTRL
 _ BINDAGENT     _ ARCHIVE       _ CREATETMTAB   _ DEBUGSESSION
 _ EXPLAIN       _ SQLADM        _ DBADM         _ DATAACCESS
 Y ACCESSCTRL    _ CREATE_SECURE_OBJECT

 FROM
   From  . . . . . . . ACCESSCTRL                                           >
 BY
   By  . . . . . . . . ____________________________________________________ >
 INCLUDING DEPENDENT PRIVILEGES
   Cascade revoke  . . ___ (Yes/No)

 Report Revoke Impacts . . . YES   (Yes/No)
 Report Dropped Synonyms & Aliases  . . NO   (Yes/No)
```

Figure 178. ***Revoke System Privileges (ADB2RZ)*** *panel*

7. Check the details on the **Revoke Impact Report (ADB2RIP)** panel.



```
ADB2RIP n -------------- DB2X Revoke Impact Report ---------------- Row 1 of 1
 Command ===> ________________________________________________ Scroll ===> PAGE



 Line commands:  I - Interpretation
                                Owner/
 S     Grantee  G Resource N/  O Schema/  Grantor/ G H Privileges/
   Lvl          T Collection   T P/K Name Binder   T G Effect
 - --- -------- - ------------ - -------- -------- - - -------------------------
 * 0   PACKADM                 Z          VNDRG      S          Y
 ******************************* END OF DB2 DATA *******************************
```

Figure 179. ***Revoke Impact Report (ADB2RIP)*** *panel*

If the following message is displayed, your user ID does not have the authority to execute the REVOKE statement:

```
Revoker does not have SYSADM/SYSCTRL/SECADM/ACCESSCTRL
```

Related tasks

“Requesting revoke impact reports in batch” on page 517
A *revoke impact report* helps you determine how the authorizations and database objects will be affected by revoking an authorization before you actually revoke it. You can request these reports on the **Revoke *object* Privileges** panels. You can also request these reports by using a batch job.

Generating SQL to re-create a Db2 object (reverse engineering)

Before making changes to a Db2 object, you might find it useful to generate the SQL statements that are required to re-create that object. Generating this SQL ensures that the changes are applied to the current definition and that the original object definitions are available for fallback purposes.

About this task

This process of generating the SQL to re-create an object is called *reverse engineering*. When requested, the reverse engineering function (also called the GEN function) of Db2 Admin Tool extracts the SQL for an object from the Db2 catalog.

In addition to extracting the DDL (data definition language) for objects, you can also generate the DCL (data control language) for all authorizations on the objects and the DML (data manipulation language) for the catalog statistics for the objects.

When you request DDL for databases, table spaces, and tables, you can request that DDL also be generated for all dependent objects, including table spaces, tables, indexes, views, synonyms, aliases, referential constraints, table check constraints, and triggers. When you request DDL for schemas, you can request that DDL also be generated for the associated distinct types, sequences, functions, global variables, stored procedures, tables, indexes, aliases, and views. Alternatively, you can specify objects that you want excluded from the generated DDL.

You can generate the SQL statements online or with a batch job. Batch jobs are recommended when you extract many objects from a large catalog.

The DDL generator function is available as a REST API. See "Provided REST APIs" on page 876.

Restrictions:

- Db2 Admin Tool does not extract IDCAMS DEFINE CLUSTER statements for VCAT-defined table spaces and indexes.
- When you reconstruct an external SQL procedure, Db2 Admin Tool cannot recover the original procedure body unless you used DSNTPSMP to create the procedure. If you used DSNTPSMP, Db2 Admin Tool can retrieve the stored procedure body from the catalog. Otherwise, if you did not use DSNTPSMP, Db2 Admin Tool cannot recreate the procedure body and replaces the original procedure body with the string "LEAVE L0". In this case, the procedure body cannot be recovered because it is not stored in the catalog. For more information about DSNTPSMP, see Creating an external SQL procedure by using DSNTPSMP (Db2 12 for z/OS).
- When the size of a native SQL procedure statement is near the 2 MB limit, sometimes Db2 Admin Tool cannot generate the DDL for the native SQL procedure statement. This situation can occur for one of the following reasons:
  - **Extra options are generated:** The native SQL procedure statement is created by GEN by first constructing the native SQL procedure options from the Db2 catalog and then appending the native SQL procedure body that is stored in the TEXT column of SYSIBM.SYSROUTINES. Sometimes the resulting DDL statement exceeds 2 MB. This situation can occur, because Db2 Admin Tool generated more options than were specified when the native SQL procedure was created. (More options can be generated when the **DB2 defaults handling** option is set to Keep.) When the 2 MB limit is exceeded, Db2 Admin Tool issues the warning message ADB1915W and generates the native SQL procedure DDL as it is stored in Db2. The resulting DDL for the native SQL procedure object is the exact contents of the TEXT column in SYSIBM.SYSROUTINES. If masking or an override was specified (such as a change owner, change schema, or Run SQLID), the ADB1916E error message is issued instead and processing stops. Processing stops, because Db2 Admin Tool cannot satisfy the 2 MB limit for the native SQL procedure DDL with the specified masks or overrides.
  - **Bytes were added for formatting:** Db2 Admin Tool attempts to format each DDL statement for readability. Sometimes during the formatting process, the extra bytes that were added for formatting cause the formatted statement length to exceed 2 MB. When this situation occurs, Db2 Admin Tool issues the ADB1919W warning message and generates unformatted DDL for the native SQL procedure. If masking or an override was specified (such as change owner, change schema, or Run SQLID), the ADB1920E error message is issued instead and processing stops. Processing stops, because Db2 Admin Tool cannot satisfy the 2 MB limit for the native SQL procedure DDL with the specified masks or overrides.

Procedure

To generate SQL to re-create a Db2 object:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify any filtering criteria at the bottom of the panel and one of the following options, and press Enter:
   - D - Databases

- S - Table spaces
- T - Tables, views, and aliases
- A - Aliases for tables and views
- Y - Synonyms
- H - Schemas
- E - User defined data types
- F - Functions
- G - Storage groups
- O - Stored procedures
- J - Triggers
- Q - Sequences and aliases
- GV - Global variables

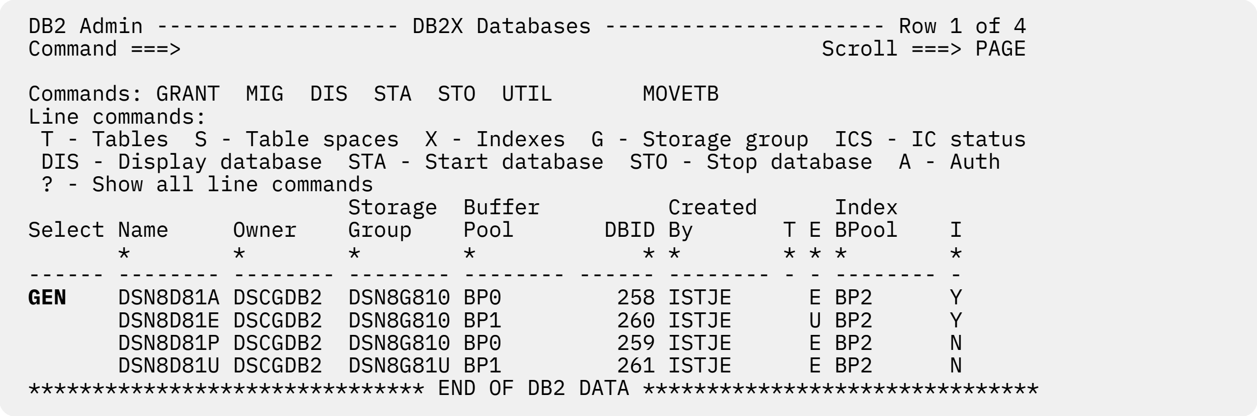
These options support the GEN command to reverse engineer objects.

3. On the relevant object panel, generate SQL by issuing one of the following commands:

   - To generate SQL for all of the listed objects, issue the GEN primary command.
   - To generate SQL for a single object, issue the GEN line command or the DDL line command against that object in the list.

     The DDL command does not provide the additional options that the GEN command provides for extracting more information, such as constraints, authorizations, or dependent objects, including triggers, labels, or comments. Instead, the DDL line command uses the default values of the GEN options. For example, pending changes are always included. For a complete list of GEN options and their default values, see "GEN options" on page 360. The DDL line command is valid everywhere that the GEN line command is valid except for the **Schemas (ADB21H)** panel (option 1.H).

   The following figure shows the GEN line command issued against a database:



```
DB2 Admin ------------------- DB2X Databases ---------------------- Row 1 of 4
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       *        *        *        *             * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
GEN    DSN8D81A DSCGDB2  DSN8G810 BP0         258 ISTJE      E BP2      Y
       DSN8D81E DSCGDB2  DSN8G810 BP1         260 ISTJE      U BP2      Y
       DSN8D81P DSCGDB2  DSN8G810 BP0         259 ISTJE      E BP2      N
       DSN8D81U DSCGDB2  DSN8G81U BP1         261 ISTJE      E BP2      N
****************************** END OF DB2 DATA *******************************
```

*Figure 180. **Databases (ADB21D)** panel - Example of issuing the GEN command to reverse engineer objects*

4. If you specified the GEN line command or primary command, complete the fields on the **Generate SQL from DB2 catalog (ADB2GENB)** panel, and press Enter.

   For more information about this panel, see "GEN options" on page 360.

   Note: For views that were created prior to DB2 9 for z/OS and for views with unqualified synonyms or aliases, you must enable the GEN function to generate SET CURRENT SQLID statements. To do so, take the following additional steps:

   a. Issue the G primary command.

b. On the resulting **Additional Generate Parameters (ADBPGENO)** panel, specify a value in the **View CURRENT SQLID method** field. This value determines which qualifier GEN is to use for the generated SET CURRENT SQLID statement if it finds multiple objects in the Db2 catalog with the unqualified name. You can specify one of the following values:

   O
   Use the qualifier of the dependent table.

   C
   Use the qualifier of the view.

   This specified value is used to set the IMPLQUALMETHOD parameter in the batch job.

5. If you requested that a batch job be generated, you can optionally edit the batch job to add wildcard characters or make any other necessary changes. Then submit the job to generate the requested SQL.

   For information about editing the batch job, see “GEN batch jobs” on page 367.

6. If you requested TSO mode, read the displayed messages and press PF3 to exit. Then take one of the following actions depending on whether you requested that the SQL be added to a WSL:

   - If you did not request that a WSL be used, the SQL statements are displayed. Review them and run them as needed by using the EXECUTE or EXECUTENF commands.
   - If you requested that a WSL be used, navigate to the WSL to view and run the statements. For instructions, see “Running a WSL” on page 551.

   Note: If when you execute these statements, the **Statement Execution Prompt (ADB2PSTM)** panel is displayed (depending on your prompt options), option 1A (runs all statements) does not apply to Change Management. If you specify option 1A and request to use Change Management [on the **Change Management Prompt (ADB2CMPR)** panel], you must register each statement individually. You need to navigate through the **Change Management Prompt (ADB2CMPR)** panel and the **Register Change (ADB2CMRG)** panel for each statement. However, you can make each statement part of the same change by specifying the same change owner and name combination for each statement.

Results

For examples of the output that is generated when you complete these steps, see “Sample output from generating SQL” on page 385.

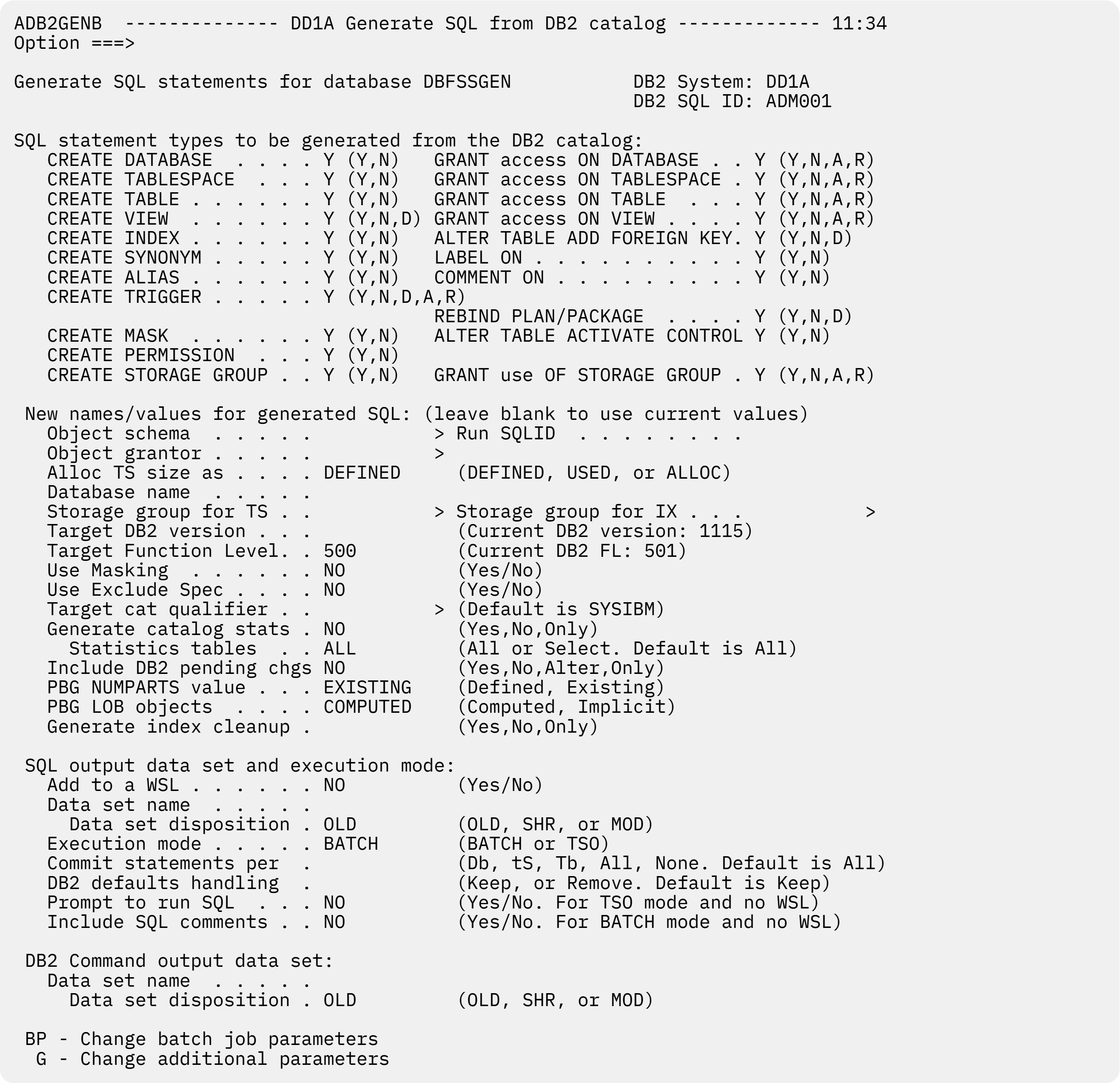
Related tasks

“Changing Db2 Admin Tool prompt options” on page 243

GEN options

When you generate the SQL for one or more objects by using the GEN function, you can specify a number of options to control the SQL that is generated. These options are available on the **Generate SQL from DB2 catalog (ADB2GENB)** panel.

The following screen shows an example of this panel when SQL is requested to recreate a database:



```
ADB2GENB  ------------- DD1A Generate SQL from DB2 catalog ------------ 11:34
Option ===>

Generate SQL statements for database DBFSSGEN           DB2 System: DD1A
                                                        DB2 SQL ID: ADM001

SQL statement types to be generated from the DB2 catalog:
   CREATE DATABASE  . . . . Y (Y,N)   GRANT access ON DATABASE . . Y (Y,N,A,R)
   CREATE TABLESPACE  . . . Y (Y,N)   GRANT access ON TABLESPACE . Y (Y,N,A,R)
   CREATE TABLE . . . . . . Y (Y,N)   GRANT access ON TABLE  . . . Y (Y,N,A,R)
   CREATE VIEW  . . . . . . Y (Y,N,D) GRANT access ON VIEW . . . . Y (Y,N,A,R)
   CREATE INDEX . . . . . . Y (Y,N)   ALTER TABLE ADD FOREIGN KEY. Y (Y,N,D)
   CREATE SYNONYM . . . . . Y (Y,N)   LABEL ON . . . . . . . . . . Y (Y,N)
   CREATE ALIAS . . . . . . Y (Y,N)   COMMENT ON . . . . . . . . . Y (Y,N)
   CREATE TRIGGER . . . . . Y (Y,N,D,A,R)
                                      REBIND PLAN/PACKAGE  . . . . Y (Y,N,D)
   CREATE MASK  . . . . . . Y (Y,N)   ALTER TABLE ACTIVATE CONTROL Y (Y,N)
   CREATE PERMISSION  . . . Y (Y,N)
   CREATE STORAGE GROUP . . Y (Y,N)   GRANT use OF STORAGE GROUP . Y (Y,N,A,R)

 New names/values for generated SQL: (leave blank to use current values)
   Object schema  . . . . .           > Run SQLID  . . . . . . . .
   Object grantor . . . . .           >
   Alloc TS size as . . . . DEFINED     (DEFINED, USED, or ALLOC)
   Database name  . . . . .
   Storage group for TS . .           > Storage group for IX . . .           >
   Target DB2 version . . .             (Current DB2 version: 1115)
   Target Function Level. . 500         (Current DB2 FL: 501)
   Use Masking  . . . . . . NO          (Yes/No)
   Use Exclude Spec . . . . NO          (Yes/No)
   Target cat qualifier . .           > (Default is SYSIBM)
   Generate catalog stats . NO          (Yes,No,Only)
     Statistics tables  . . ALL         (All or Select. Default is All)
   Include DB2 pending chgs NO          (Yes,No,Alter,Only)
   PBG NUMPARTS value . . . EXISTING    (Defined, Existing)
   PBG LOB objects  . . . . COMPUTED    (Computed, Implicit)
   Generate index cleanup .             (Yes,No,Only)

 SQL output data set and execution mode:
   Add to a WSL . . . . . . NO          (Yes/No)
   Data set name  . . . . .
     Data set disposition . OLD         (OLD, SHR, or MOD)
   Execution mode . . . . . BATCH       (BATCH or TSO)
   Commit statements per  .             (Db, tS, Tb, All, None. Default is All)
   DB2 defaults handling  .             (Keep, or Remove. Default is Keep)
   Prompt to run SQL  . . . NO          (Yes/No. For TSO mode and no WSL)
   Include SQL comments . . NO          (Yes/No. For BATCH mode and no WSL)

 DB2 Command output data set:
   Data set name  . . . . .
     Data set disposition . OLD         (OLD, SHR, or MOD)

 BP - Change batch job parameters
  G - Change additional parameters
```

Figure 181. ***Generate SQL from DB2 catalog (ADB2GENB)*** *panel for a database*

This panel contains the following fields:

- The first set of fields specify whether SQL statements are to be generated for the requested objects and any dependent objects, where applicable. The fields that are displayed depend on the type of object for which you requested SQL:

  CREATE DATABASE

  Specifies whether CREATE statements are to be generated for all of the explicitly requested databases.

  When you also request to generate storage groups, statements are generated for the default storage group.

  CREATE TABLESPACE

  Specifies whether CREATE statements are to be generated for all of the table spaces that are identified during processing, which includes both explicitly and implicitly requested table spaces. For example, if you specify the GEN command for a database and specify Y in the **CREATE**

TABLESPACE field, a CREATE statement is generated for each table space that resides in the database.

CREATE parent DATABASE

Specifies whether CREATE statements are to be generated for the database that contains the table or table space.

This field is applicable to only tables and table spaces. In the case of a table, if Y is specified, **CREATE parent TABLESPACE** is implicitly set to Y, regardless of the value specified for that field. This behavior ensures the generated DDL does not fail due to a missing CREATE TABLESPACE statement.

CREATE parent TABLESPACE

Specifies whether CREATE statements are to be generated for the table space that contains the table. This field is applicable to only tables.

CREATE TABLE

Specifies whether CREATE statements are to be generated for all of the tables that are identified during processing, which includes both explicitly and implicitly requested tables.

CREATE VIEW

Specifies whether CREATE statements are to be generated for all of the views that are identified during processing, which includes both explicitly and implicitly requested views.

Specify D to extract views without requiring Db2 Admin Tool to check whether all other objects that used in the view are also being generated. This option significantly reduces the resource consumption when running on large Db2 catalogs.

CREATE INDEX

Specifies whether CREATE statements are to be generated for all of the indexes that are identified during processing, which includes both explicitly and implicitly requested indexes.

CREATE SYNONYM

Specifies whether CREATE statements are to be generated for all of the synonyms that are identified during processing, which includes both explicitly and implicitly requested synonyms.

CREATE ALIAS

Specifies whether CREATE statements are to be generated for all of the aliases that are identified during processing, which includes both explicitly and implicitly requested aliases.

CREATE TRIGGER

Specifies whether CREATE statements are to be generated for all of the triggers that are identified during processing, which includes both explicitly and implicitly requested triggers.

CREATE MASK

Specifies whether CREATE statements are to be generated for all of the masks that are identified during processing, which includes both explicit and implicit masks.

CREATE PERMISSION

Specifies whether CREATE statements are to be generated for all of the permissions that are identified during processing, which includes both explicit and implicit permissions.

CREATE STORAGE GROUP

Specifies whether CREATE statements are to be generated for all of the storage groups that are identified during processing, which includes both explicit and implicit storage groups.

GRANT access ON DATABASE

Specifies whether a GRANT *access* ON DATABASE statement is to be generated.

The following values are valid for this field and the other GRANT fields:

Y

Generate GRANT statements for authorizations and roles.

N

Do not generate any GRANT statements.

A

Generate GRANT statements for authorizations.

R
Generate GRANT statements for roles.

GRANT access ON TABLESPACE
Specifies whether a GRANT *access* ON TABLESPACE statement is to be generated. Valid values are listed in "GRANT access ON DATABASE" on page 362.

GRANT access ON TABLE
Specifies whether a GRANT *access* ON TABLE statement is to be generated. Valid values are listed in "GRANT access ON DATABASE" on page 362.

GRANT access ON VIEW
Specifies whether a GRANT *access* ON VIEW statement is to be generated. Valid values are listed in "GRANT access ON DATABASE" on page 362.

ALTER TABLE ADD FOREIGN KEY
Specifies whether an ALTER TABLE ADD FOREIGN KEY statement is to be generated. Specify D to extract foreign keys for tables that are dependent on the tables that are extracted.

LABEL ON
Specifies whether a LABEL ON statement is to be generated.

COMMENT ON
Specifies whether a COMMENT ON statement is to be generated.

REBIND PLAN/PACKAGE
Specifies whether REBIND commands for plans and packages are to be generated. These REBIND commands are written to the data set that is specified in the **DB2 Command output file: Data set name** field.

ALTER TABLE ACTIVATE CONTROL
Activates an enabled masked column. A column mask can be created as enabled or disabled for column access control. An enabled column mask does not take effect until the ALTER TABLE statement with the ACTIVATE COLUMN ACCESS CONTROL clause is used to activate column access control for the table.

GRANT use OF STORAGE GROUP
Specifies whether a GRANT USE OF STOGROUP statement is to be generated. Valid values are listed in "GRANT access ON DATABASE" on page 362.

- The second set of fields specify the new names or values to be used in the generated SQL:

Object schema
Specifies a schema to use for any new objects.

Run SQLID
Specifies the SQL ID to use when creating objects. The SQL ID that is specified must have the privileges that are necessary to create objects, such as an administrative type of SQL ID that has been defined. If you specify a value of <NONE>, no SET CURRENT SQLID statements are generated in the DDL. If you leave the field blank, a SET CURRENT SQLID statement is generated in the DDL before each object that is created; where possible, the SQL ID that was originally used to create the object is used.

If you specify an SQL ID of <NONE> and use synonyms, the following behaviors apply:

- If the creator of the synonym is the same as the creator of the table on which the synonym is defined, an executable CREATE SYNONYM statement is generated.
- If the creator of the synonym is not the same as the creator of the table on which the synonym is defined, the SQL ID that created the synonym is extracted from the catalog and both the SET SQLID and CREATE SYNONYM statements are created. However, these statements are commented out, and an informational message is issued. Because these statements are included in comments, other generated statements might fail. For example, a view that is defined using the synonym might fail.

The other Db2 Admin Tool functions where you can specify a RUN SQLID value include the Rename Database, ALT, Migrate, and Change Management functions.

Object grantor

Specifies the grantor of the object.

Alloc TS size as

Specifies how to generate the primary quantity. The following values are valid:

DEFINED

Uses the size that is defined in the catalog.

USED

Uses the size that is actually used.

ALLOC

Uses the allocated size.

If you specify USED or ALLOC, ensure that you have recently run the RUNSTATS utility on the selected objects and the STOSPACE utility on the storage groups for the objects that are generated. Running these utilities is necessary, because Db2 Admin Tool depends on information in the Db2 catalog to generate actual allocated space or actual used space. The actual data set sizes for table spaces and index spaces are not retrieved.

Database name

Specifies a new database name for the objects (except when initiated by using a primary command from a list of databases).

Storage group for TS

Specifies a new storage group for the table spaces.

Storage group for IX

Specifies a new storage group for the indexes.

Target DB2 version

Specifies the Db2 level for the generated SQL statements, if different from the current Db2 level.

The Db2 level format is *vvrm*, where *vv*=version, *r*=release, and *m*=modification level. The current Db2 level is the default.

Sometimes SQL syntax support is removed from Db2. Specifying the correct target Db2 version ensures that the generated SQL is valid for the target Db2 subsystem. For example, for CREATE INDEX, Db2 11 supports EXCLUDE NULL KEYS, but Db2 10 does not.

The following values are examples of valid level values:

1010

Db2 10 compatibility mode (CM8)

1012

Db2 10 compatibility mode (CM9)

1013

Db2 10 enabling-new-function mode (ENFM)

1015

Db2 10 new-function mode (NFM)

1110

Db2 11 conversion mode (CM)

1113

Db2 11 enabling-new-function mode (ENFM)

1115

Db2 11 new-function mode (NFM)

Example: Suppose that your current Db2 level is Db2 11 new-function mode, but you want to generate SQL that runs on a Db2 10 new-function mode system. Set 1015 as the target Db2 version.

The IN DD run parameter DB2REL uses the same format and values as the **Target DB2 version** option. When Db2 Admin Tool generates a GEN batch job, it gets the Db2 release level from an SQL

CONNECT statement and uses that release level value in the generated job. Use this generated job as the base for defining customized GEN jobs.

Target Function Level
Specifies the target function level. DDL is generated based on the syntax requirements for the specified target function level. Valid values are any integer value in the range 501 - 999 or 100. You can specify any function level equal to or less than the current Db2 version function level.

Include DB2 pending chgs
Specifies how to include Db2 pending changes. Valid values are:

`Yes`
Include the Db2 pending changes in the generated CREATE statements for table spaces and indexes. `Yes` is the default.

`No`
Generate SQL comments that contain ALTER statements for the Db2 pending changes. The Db2 pending changes are not included in the generated CREATE statements for table spaces and indexes.

`Alter`
Generate ALTER statements for the Db2 pending changes.

`Only`
Generate only ALTER statements for the Db2 pending changes. No other SQL statements, such as CREATE statements, are generated.

PBG NUMPARTS value
Specifies the value for the NUMPARTS clause when re-creating a partition-by-growth (PBG) table space. Valid values are:

`Defined`
Use the NUMPARTS value that was specified when the table space was created.

`Existing`
Use the existing NUMPARTS value. This existing value includes any added partitions. This value can be different from the value that was defined when the table space was created. `Existing` is the default.

PBG LOB objects
Specifies whether the auxiliary objects for LOB columns in a partition-by-growth (PBG) table space are to be re-created implicitly or explicitly. Valid values are:

`Computed`
The auxiliary objects are to be re-created explicitly if all of the required auxiliary objects exist and were created explicitly. `Computed` is the default.

`Implicit`
The auxiliary objects are to be re-created implicitly by Db2.

Generate index cleanup
Specifies index cleanup options. Valid values are:

`Yes`
Generate DML statements for the Db2 SYSINDEXCLEANUP catalog table.

`No`
Do not generate DML statements for the Db2 SYSINDEXCLEANUP catalog table. No is the default value.

`Only`
Generate DML statements only for the Db2 SYSINDEXCLEANUP catalog table. No other DDL statements, such as CREATE statements, or DML statements, such as statements for catalog statistics, are to be generated.

Use Masking
Specifies whether to enable masking. For instructions on how to specify masks, see "Specifying masks" on page 315.

Use Exclude Spec
Specifies whether to use an existing exclude specification. An *exclude specification* enables you to select objects to exclude from the generated DDL.

Target cat qualifier
Specifies the qualifier to use in the INSERT, UPDATE, and DELETE statements for updating catalog statistics and for index cleanup settings.

Generate catalog stats
Specifies whether to generate catalog statistics, which means that INSERT, UPDATE, and DELETE statements that modify the catalog statistics are included in the DDL file. Valid value are:

Y
Generate DDL and catalog statistics.

N
Generate DDL only. Do not generate catalog statistics.

O
Generate catalog statistics only. Do not generate DDL.

The statistic fields that are modified are those fields that are associated with the objects that are generated. (The complete list of statistics fields are those fields that are set by RUNSTATS and that can be modified and the statistics columns for table functions in SYSROUTINES, which are not set by RUNSTATS.)

Statistics tables
Specifies which statistics to generate. Valid values are:

All
Generate all statistics. All is the default.

Select
Generate selected statistics. If you specify Select, the **Catalog Statistics Tables (ADBPGEN2)** panel opens. On this panel, you can select the catalog tables for which you want to generate DML statements.

- The third set of fields specify the output file and execution mode options:

Add to work stmt list
Specifies whether to save the output to a work statement list (WSL) data set.

Data set name
Specifies the name of the data set in which to place the generated SQL. The value must be a valid SPUFI input data set name or SYSOUT=x. The default is SYSOUT=*. If you leave this field blank, the command output is created as comments in the output file.

Data set disposition
Specifies the disposition of the output data set.

Execution mode
Specifies the execution mode for generating the SQL. Valid values are:

BATCH
Runs the SQL generation as a batch job. Db2 Admin Tool generates the batch job and displays the job in an ISPF edit session, where you can make any changes before submitting the job for execution.

TSO
Runs the SQL generation online. Db2 Admin Tool generates the SQL statements online and displays the results.

Commit statements per
Specifies how often an SQL COMMIT statement is added to the generated SQL. Valid values are:

D
COMMIT statements are run for each database.

S
COMMIT statements are run for each table space.

T
COMMIT statements are run for each table.

A
COMMIT statements are run for all objects. A is the default.

N
COMMIT statements are never run.

The exception is if the generated SQL includes 500 or more statements that update catalog statistics. In this case, even if you specify N, COMMIT statements are added to avoid catalog locking problems. A COMMIT statement is added at least every 500 statements that modify statistics.

DB2 defaults handling
Specifies whether Db2 default parameters are kept in the generated SQL. Valid values are:

K
Keeps the default parameters. K is the default.

R
Removes the default parameters.

Prompt to run SQL
Specifies whether, after the SQL edit session, Db2 Admin Tool is to display a prompt that allows you to choose whether to run the SQL immediately. This option applies only when you are using TSO mode without WSL. Valid values are:

Y
Display this prompt.

The maximum number of SQL statements that are allowed is 8120. The maximum length of an SQL statement is 2,097,152 bytes (2 MB).

N
Do not display this prompt. N is the default.

- The last set of fields specify the following options for the command output file:

Data set name
Specifies the name of the data set in which to place the generated REBIND commands if **REBIND PLAN/PACKAGE** is set to Y or D.

Data set disposition
Specifies the disposition of the output data set.

GEN batch jobs

When you use the reverse engineering function (GEN) to generate SQL, you have the option to request batch as the execution mode. If you specify this option, Db2 Admin Tool generates a JCL batch job. You can optionally edit this job to use wildcard characters in the qualifiers and names of the objects. Using wildcard characters enables the DDL to be extracted for objects based on string patterns in the object name.

The following JCL is an example of one of these batch jobs:

```
//GENSQL   EXEC PGM=IKJEFT01,DYNAMNBR=100
//STEPLIB  DD DISP=SHR,DSN=ADBC10.ISPLLIB
//         DD DISP=SHR,DSN=DSN.DSNA.SDSNEXIT
//         DD DISP=SHR,DSN=DSN.DSNA.SDSNLOAD
//         DD DISP=SHR,DSN=AUTHSW.ISPLLIB
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  DSN SYSTEM(DSNA)
  RUN PROG(ADB2GEN) PLAN(ADB) PARMS('/REBIND')
  END
/*
//SYSPRINT DD SYSOUT=*
//SQL      DD SYSOUT=*,DCB=(RECFM=FB,LRECL=80)
//IN       DD *
  DB2SYS   = 'DSNA',
  DB2ALOC  = '',
  DB2SERV  = 'DSNA',
  DB2AUTH  = 'SINNOTT',
  DB2REL   = '1013',
  GENSG    = 'Y',
  GENDB    = 'Y',
  GENTS    = 'Y',
  GENTABLE = 'Y',
  GENVIEW  = 'Y',
.
.
.
  NEWGRANTOR = '',
  SPCALLOC = 'DEFINED',
  TGTDB2   = '';
  TYPE='DB',QUAL='',NAME='DSNDB04';
```

Figure 182. Example JCL to generate SQL

The in-stream data set identified by the IN DD statement contains both run parameters and request parameters:

Run parameters

Specify options for how the SQL is to be generated. They are listed first in the IN data set. In the preceding example, the semicolon (;) after the TGTDB2 parameter indicates the end of the list of run parameters. The next line contains request parameters.

Restriction: You cannot modify run parameters in the generated batch job.

Request parameters

Name the specific objects for which SQL is to be generated. The request parameters must follow the run parameters in the IN data set. The request parameters are:

- TYPE
- QUAL
- NAME
- VERSION
- INCLUDE
- XDTYPE

The TYPE, QUAL, and NAME parameters together identify an object or objects by name. If the object is a native stored procedure, user-defined function, or trigger, the optional VERSION parameter indicates a particular version. If the object is a table or table space, the optional INCLUDE parameter indicates whether to include parent objects. See "Request parameters: TYPE, QUAL, and NAME" on page 369. You can also optionally use the XDTYPE parameter to exclude objects. See "Optional XDTYPE parameter" on page 370.

Wildcard characters in request parameters

You can modify the request parameters in the generated JCL with wildcard characters as needed. The values for the qualifier and name can contain one or more of the following wildcard characters:

-

A minus sign (-) represents any single character.

% or *

A percent sign (%) or asterisk (*) represents one or more characters.

Any other character represents a single occurrence of itself.

The rules for wildcard characters are the same as the rules for the LIKE predicate.

Request parameters: TYPE, QUAL, and NAME

The combination of TYPE, QUAL, and NAME parameters specify the request for an object. For example, the following request generates the DDL for database DSNDB04 and all of the objects that it contains:

```
TYPE='DB',QUAL='',NAME='DSNDB04';
```

The following table shows the values that can be specified for the request parameters TYPE, QUAL, and NAME for each type of object.

Table 22. Values of TYPE, QUAL, and NAME for each object type

Object Type	TYPE	QUAL	NAME
Database	DB	n/a	*dbanme*
Table space[1]	TS	*dbname*	*tsname*
Table[1]	TB	*creator*	*tbname*
Global Variable	GV	*schema*	*gvname*
View	VW	*creator*	*vwname*
Alias	AL	*creator*	*aliasname*
Index	IX	*creator*	*ixname*
User-defined data type	DT	*schema*	*udtname*
User-defined function	FU	*schema*	*udfname*
Stored procedure[2]	SP	*schema*	*stpname*
Sequence	SQ	*schema*	*seqname*
Schema	SC	*schema*	n/a
Extended Schema	XS	*schema*	n/a
Trigger	TG	*schema*	*tgname*
Storage group	SG	n/a	*sgname*
Synonym	SY	*creator*	*syname*
Trusted context	TC	n/a	*tcname*
Role	RO	n/a	*roname*

Notes:

1. For tables and table spaces, you can also specify an INCLUDE parameter to specify that parent objects are to be included. For details, see "Optional INCLUDE parameter" on page 370.
2. For native SQL procedures, user-defined functions, and triggers, you can also specify a VERSION parameter. For details, see "Optional VERSION parameter" on page 370.

Optional INCLUDE parameter

For tables and table spaces only, you can also specify an INCLUDE parameter as follows to include parent objects in the generated DDL:

- For a table, you can generate the SQL for the parent table space (TS) and parent database (DB). Specify either TS or DB or both.
- For a table space, you can generate the SQL for the parent database (DB). You can specify only DB for table spaces.

You cannot specify wildcard characters for the INCLUDE parameter. For syntax details, see the syntax for request-list.

For example, the following requests specify that the parent objects be included:

```
TYPE='TS',QUAL='*',NAME='TS10*',INCLUDE='DB';

TYPE='TB',QUAL='TS3071',NAME='TB*',INCLUDE='DB,TS';
```

Optional XDTYPE parameter

You can specify the XDTYPE parameter to exclude objects. For example, the following request is used in the MIG process on the database level:

```
XDTYPE ='TS',QUAL='BFVDCH',NAME='BFVSCHK';

TYPE='DB',NAME='BFVDCH';
```

These two requests specify all table spaces in database BFVDCH, except for table space BFVSCHK.

Optional VERSION parameter

For native SQL procedures, user-defined functions, and triggers, you can also specify a VERSION parameter after TYPE, QUAL, and NAME to indicate the version or versions for which you want to extract the DDL. You can specify a specific version, the active version, or all versions. For example, the following request specifies version V1:

```
TYPE='SP',QUAL='DEMBIN2',NAME='MYSTP',VERSION='V1';
```

Wildcard characters can also be used in the VERSION parameter. For example, the following specification extracts DDL for all versions:

```
VERSION='*'
```

If the version is omitted or set to blank, the active version is extracted. For example, the following specification extracts DDL for all active stored procedures within the TEST schema:

```
QUAL='TEST',NAME='*'
```

ADB2RE stored procedure

The ADB2RE stored procedure generates SQL for objects from the Db2 catalog.

The input parameters identify the object types for which to generate SQL, a set of requests that specify the starting points for the objects, and the type of result sets that ADB2RE is to return.

The ADB2RE stored procedure can return two result sets: the SQL and the report. The result sets can be saved in data sets on the system where ADB2RE runs.

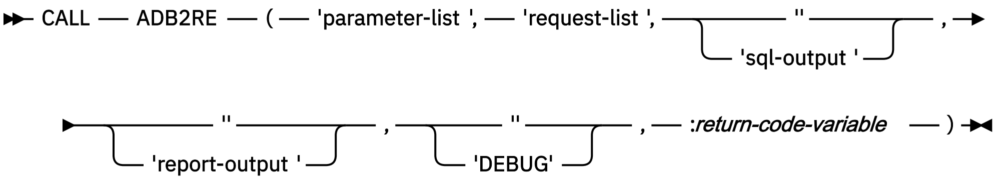
Environment

For information about defining the ADB2RE stored procedure, see “Defining the provided stored procedures ” on page 117. The ADB2RE stored procedure runs in a WLM-established stored procedures address space.

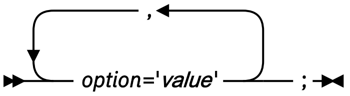
Authorization

To execute the CALL statement, the owner of the package or plan that contains the CALL statement must have EXECUTE privilege on the ADB2RE stored procedure.

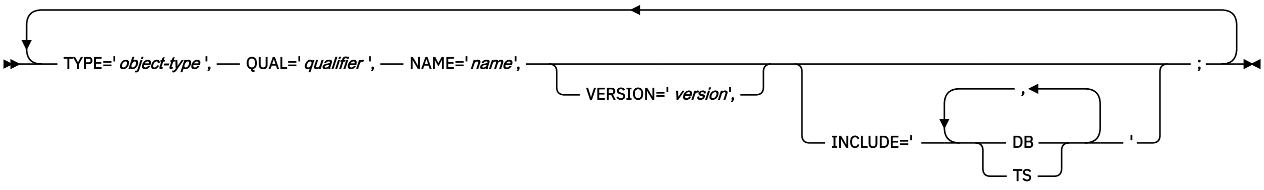
Syntax



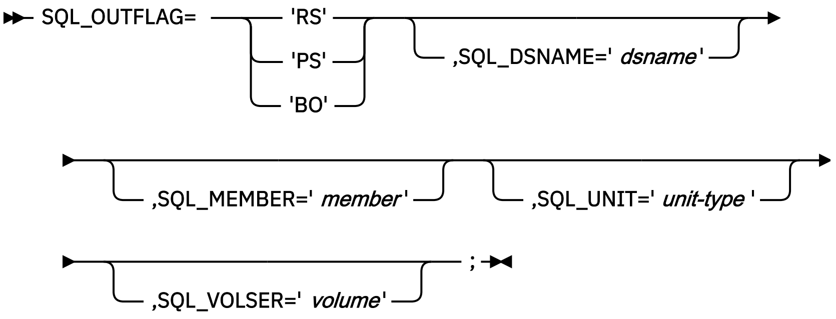
parameter-list



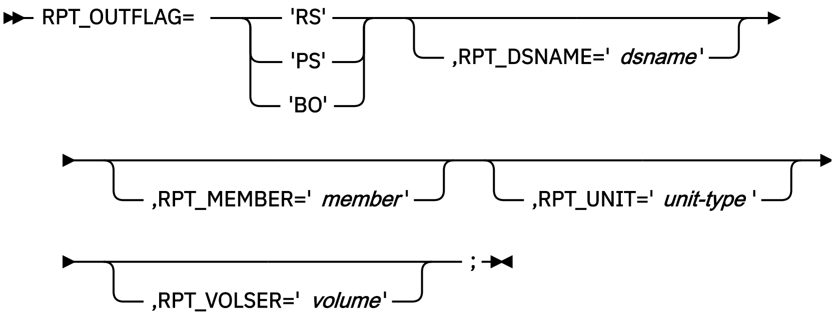
request-list



sql-output



report-output



Option descriptions

parameter-list

Lists the input parameters for ADB2RE. *parameter-list* is VARCHAR(500) in EBCDIC and is required. *parameter-list* is specified in the format: `option-1='value-1', option-2='value-2', … option-n='value-n'`. The following table lists the options for *parameter-list*.

Table 23. Options for parameter-list

Option	Allowed values	Description
General Db2 options		
DB2SYS	*db2-ssid*	The current Db2 subsystem identifier
DB2ALOC	*location-name*	The location name of the Db2 subsystem to connect to
DB2SERV	*current-location-name*	The name of the current location
DB2AUTH	*current-sqlid*	The current SQLID
DB2REL	*db2-lvl*	The current release level of the Db2 subsystem
Options to generate DDL statements		
GENSG	Y or N	A value of Y specifies that CREATE statements for all of the storage groups that are identified during processing are to be generated, which includes both explicit and implicit storage groups.
GENDB	Y or N	A value of Y specifies that CREATE statements for all of the explicitly requested databases are to be generated. When you also request to generate storage groups, statements are generated for the default storage group.
GENTS	Y or N	A value of Y specifies that CREATE statements for all of the table spaces that are identified during processing are to be generated, which includes both explicitly and implicitly defined table spaces. For example, if you specify the GENDB='Y' and specify GENTS='Y', a CREATE statement will be generated for each table space that resides in the database.

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
GENTABLE	Y or N	A value of Y specifies that CREATE statements for all of the tables that are identified during processing are to be generated, which includes both explicitly and implicitly defined tables.
GENVIEW	Y, N, or D	A value of Y specifies that CREATE statements for all of the views that are identified during processing are to be generated, which includes both explicitly and implicitly defined views. Specify D to extract views without Db2 Admin Tool checking whether all other objects used in the view are also being generated. This option significantly reduces the resource consumption when running on large Db2 catalogs.
GENINDEX	Y or N	A value of Y specifies that CREATE statements for all of the indexes that are identified during processing are to be generated, which includes both explicitly and implicitly defined indexes.
GENSYN	Y or N	A value of Y specifies that CREATE statements for all of the synonyms that are identified during processing are to be generated, which includes both explicitly and implicitly defined synonyms.
GENALIAS	Y or N	A value of Y specifies that CREATE statements for all of the aliases that are identified during processing are to be generated, which includes both explicitly and implicitly defined aliases.
GENUDT	Y or N	A value of Y specifies that CREATE statements for all of the user-defined types that are identified during processing are to be generated.
GENUDF	Y or N	A value of Y specifies that CREATE statements for all of the user-defined functions that are identified during processing are to be generated.

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
GENSTP	Y or N	A value of Y specifies that CREATE statements for all of the stored procedures that are identified during processing are to be generated.
GENSEQ	Y or N	A value of Y specifies that CREATE statements for all of the sequences that are identified during processing are to be generated.
GENVAR	Y or N	A value of Y specifies that CREATE statements for all of the global variables that are identified during processing are to be generated.
GENLABEL	Y or N	Generates a LABEL ON statement in the SQL.
GENCOMM	Y or N	Generates a COMMENT ON statement in the SQL.
GENRELS	Y, N, or D	Generates DDL for ALTER TABLE ADD FOREIGN KEY. Specify D to extract FOREIGN KEYS for tables that are dependent on the tables being extracted.
GENTRIG	Y, N, or D	A value of Y specifies that CREATE statements for all of the triggers that are identified during processing are to be generated, which includes both explicitly and implicitly defined triggers.
GENTRUST	Y or N	A value of Y specifies that CREATE statements for all of the trusted contexts that are identified during processing are to be generated, which includes both explicit and implicit trusted contexts.
GENROLE	Y or N	A value of Y specifies that CREATE statements for all of the roles that are identified during processing are to be generated, which includes both explicit and implicit roles.

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
GENMASK	Y or N	A value of Y specifies that CREATE statements for all of the masks that are identified during processing are to be generated, which includes both explicit and implicit masks.
GENPERM	Y or N	A value of Y specifies that CREATE statements for all of the permissions that are identified during processing are to be generated, which includes both explicit and implicit permissions.
Options to generate GRANT statements		
GRANTSG	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT USE OF STOGROUP statement in the SQL.
GRANTDB	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT access ON DATABASE statement in the SQL.
GRANTTS	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT access ON TABLESPACE statement in the SQL.

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
GRANTTAB	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT access ON TABLE statement in the SQL.
GRANTVW	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT access ON VIEW statement in the SQL.
GRANTSCH	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT access ON SCHEMA statement in the SQL.
GRANTUDT	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT USAGE ON user-defined type statement in the SQL.

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
GRANTUDF	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT EXECUTE ON user-defined function statement in the SQL.
GRANTSTP	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT EXECUTE ON stored procedure statement in the SQL.
GRANTSEQ	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT access ON SEQUENCE statement in the SQL.
GRANTVAR	**Y** Generate GRANT statements for authorizations and roles **N** Do not generate any GRANT statements **A** Generate GRANT statements for authorizations **R** Generate GRANT statements for roles	Generates a GRANT access ON VARIABLE statement in the SQL.
Additional options to generate statements		

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
ACCEPT_FL	An integer value in the range 501 - 999.	Specify the maximum Db2 version function level (FL) with which the stored procedure is allowed to continue. If the specified value is greater than the function level that is supported by Db2 Admin Tool, a warning message is issued. If the specified value is less than the current function level value of the Db2 subsystem, an error message is issued and the job is terminated.
ACTVCNTL	Y or N	Activates an enabled masked column. A column mask can be created as enabled or disabled for column access control. An enabled column mask does not take effect until the ALTER TABLE statement with the ACTIVATE COLUMN ACCESS CONTROL clause is used to activate column access control for the table.
CATALOGSTATISTICS	**Y** Generate DDL and catalog statistics. **N** Generate DDL only. Do not generate catalog statistics. **O** Generate catalog statistics only. Do not generate DDL.	Specify whether to generate catalog statistics, which causes INSERT, UPDATE, and DELETE statements that modify the catalog statistics to be included in the DDL file. The statistic fields that are generated are those that are associated with the objects that are being generated. (The complete list of statistics fields are those fields that are set by RUNSTATS that can be modified and the five statistics columns for table functions in SYSROUTINES, which are not set by RUNSTATS.)
SQLCMTS	**Y** Include SQL comments **N** Do not include SQL comments. N is the default value.	Controls whether SQL comments within a native SQL procedure, SQL function, or view are generated. This option applies to only native SQL procedures, SQL functions, and views.

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
TCATQUAL	*schema-name*	Specify the qualifier to be used in the INSERT, UPDATE, and DELETE statements for updating catalog statistics and for index cleanup settings.
TGTFL	An integer value in the range 501 - 999 or 100. You can specify any function level equal to or less than the current Db2 version function level.	Specify the target function level. DDL statements are generated based on the syntax requirements for the specified target function level.
Options to change DDL during generation[1]		
NEWSCH	*schema-name*	Specify a new object schema. If specified, the new schema is used whenever an object is created.
NEWGRANTOR	*grantor-name*	The grantor of the object.
NEWDB	*database-name*	Specify a new database name for the objects (except when initiated using a primary command from a list of databases).
NEWTSSG	*stogroup-name*	Specify a new storage group for the table spaces.
NEWIXSG	*stogroup-name*	Specify a new storage group for the indexes.
NEWSQLID	*sqlid*	The SQLID to use as the new owner for the object of the generated DDL.
Additional options to customize the DDL		

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
PENDCHGS	**Y** (default) Include the Db2 pending changes when generating CREATE statements for table spaces and indexes. **N** Generate SQL comments that contain ALTER statements for the Db2 pending changes. The Db2 pending changes are not included when generating CREATE statements for table spaces and indexes. **A** Generate ALTER statements for the Db2 pending changes. **O** Only generate ALTER statements for the Db2 pending changes. No other SQL (such as CREATE statements) will be generated.	Specify additional methods of including Db2 pending changes.
SPCALLOC	**DEFINED** Uses the size defined in the catalog. **ALLOC** Uses the allocated size. **USED** Uses the size that is actually used.	Specifies how to generate the primary quantity. Specifying either the ALLOC or USED values requires you to run the STOSPACE utility for the storage groups for the objects being generated.

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
TGTDB2	*vvrm*	Specify the Db2 level for the generated SQL statements, if different from the current Db2 level. The Db2 level format is *vvrm*, where *vv*=version, *r*=release, and *m*=modification level. The current Db2 level is the default. **Important:** Sometimes SQL syntax support is removed from Db2. Specifying the correct target Db2 version ensures that the generated SQL will be valid for the target Db2 subsystem. For example, for CREATE INDEX, Db2 11 supports EXCLUDE NULL KEYS, but Db2 10 does not. The following values are examples of valid level values: **1110** Db2 11 compatibility mode (CM) **1113** Db2 11 enabling NFM **1115** Db2 11 NFM **1010** Db2 10 compatibility mode (CM8) **1012** Db2 10 compatibility mode (CM9) **1013** Db2 10 enabling NFM **1015** Db2 10 NFM **Example:** Suppose that your current Db2 level is Db2 11 NFM (1115), but you want to generate SQL that runs on a Db2 10 NFM system. Set 1015 as the target Db2 version.

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
DEFAULTS	**K** Keeps Db2 default parameters (default). **R** Removes Db2 default parameters.	Specify whether Db2 default parameters should be removed or kept in the generated SQL
COMMITFR	**A** Commit statements are run for all objects (default). **D** Commit statements are run for each database. **S** Commit statements are run for each table space. **T** Commit statements are run for each table. **N** Commit statements are never run.	Specify how often an SQL COMMIT statement is added to the generated SQL.

Table 23. Options for parameter-list (continued)

Option	Allowed values	Description
RUNSQLID	*sqlid*	Specify the SQL ID to be used when creating objects. The SQL ID that is specified must have the privileges that are necessary to create objects, such as an administrative type of SQL ID that has been defined. If you specify a value of <NONE>, no SET CURRENT SQLID statements are generated in the DDL. If you leave the field blank, a SET CURRENT SQLID statement is generated in the DDL before each object that is created (where possible, the SQL ID that was originally used to create the object is used). If you specify an SQLID of <NONE>, the following is true if you use synonyms: • If the creator of the synonym is the same as the creator of the table on which the synonym is defined, an executable CREATE SYNONYM statement is generated. • If the creator of the synonym is not the same as the creator of the table on which the synonym is defined, the SQLID that created the SYNONYM is extracted from the catalog and both the SET SQLID and CREATE SYNONYM statements are created, but commented out. An informational message is issued. Be aware that other generated statements might fail due to these statements being commented out (for example, a view that is defined using the synonym).

Note:

1. Replacing IDs and names in the DDL is useful when cloning objects on the same Db2 subsystem.

request-list

Lists the object requests for ADB2RE. *request-list* is VARCHAR(4000) in EBCDIC and is required. *request-list* is specified in the format: `request-1;request-2; … request-n;`.

A single request is formatted as follows:

```
TYPE='object-type',QUAL='qualifier',NAME='name'[,VERSION='version'][,INCLUDE='object-type'];
```

For example, to request DDL for database DSNDB04 (including for all dependent objects depending on the options specified in *parameter-list*), the request would look similar to the following:

```
TYPE='DB',QUAL='',NAME='DSNDB04';
```

Requests are described in detail in "GEN batch jobs" on page 367.

sql-output

Specifies the type of SQL output to return and the data set to use. *sql-output* is VARCHAR(120) in EBCDIC and is optional. If *sql-output* is not specified, a null string ('') must be specified.

SQL_OUTFLAG='RS'|'PS'|'BO'

Identifies the type of SQL output to return. RS returns a result set. PS returns a data set. BO returns both a result set and a data set.

SQL_DSNAME='*dsname*'

Optionally identifies a data set name. If SQL_DSNAME is not specified, a temporary data set will be used for SQL output.

SQL_MEMBER='*member*'

Optionally identifies a member name.

SQL_UNIT='*unit-type*'

Optionally identifies a group name. If SQL_UNIT is not specified, SYSALLDA is used for the group name.

SQL_VOLSER='*volume*'

Optionally identifies a volume name.

report-output

Specifies the type of report output to return and the data set to use. *report-output* is VARCHAR(120) in EBCDIC and is optional. If *report-output* is not specified, a null string ('') must be specified.

RPT_OUTFLAG='RS'|'PS'|'BO'

Identifies the type of report output to return. RS returns a result set. PS returns a data set. BO returns both a result set and a data set.

RPT_DSNAME='*dsname*'

Optionally identifies a data set name. If RPT_DSNAME is not specified, a temporary data set will be used for report output.

RPT_MEMBER='*member*'

Optionally identifies a member name.

RPT_UNIT='*unit-type*'

Optionally identifies a group name. If RPT_UNIT is not specified, SYSALLDA is used for the group name.

RPT_VOLSER='*volume*'

Optionally identifies a volume name.

'' or 'DEBUG'

Specifies whether to use debug processing. To activate debug processing, specify 'DEBUG'. If DEBUG is not specified, a null string ('') must be specified.

:*return-code-variable*

A Db2 for z/OS variable that is used for any return code generated by ADB2RE. The variable is SMALLINT and is an output parameter. ADB2RE will return one of the following return codes:

0 - Successful
4 - Successful with warning messages issued
8 - Error(s) in specified parameters
12 - Processing error

16 - Error in dynamic allocation of the DDL data set (sql-output)

20 - Error in dynamic allocation of the report data set (report-output)

Result set formats

SQL result set

The SQL result set uses a cursor named C_SQL and contains the following columns:

Column	Description
SEQ INTEGER NOT NULL	The sequence number in the SQL
LINE CHAR(80) EBCDIC	The SQL line output

Report result set

The report result set uses a cursor named C_RPT and contains the following columns:

Column	Description
SEQ INTEGER NOT NULL	The sequence number in the report
LINE VARCHAR(133) EBCDIC	The report line

Example

Issue the following CALL statement to generate all CREATE TABLE and CREATE INDEX statements for the Db2 catalog (database DSNDB06), return result sets for the SQL and the report, and return a return code. The CREATE DATABASE and CREATE TABLESPACE statements are not returned because 'gendb' and 'gents' input options are specified with 'N'.

```
CALL ADB2RE('db2sys=''DSNA'',gendb=''N'',gents=''N'',gentable=''Y'',genindex=''Y'';',
            'type=''DB'',name=''DSNDB06'';',
            'SQL_OUTFLAG=''RS'';',
            'RPT_OUTFLAG=''RS'';',
            '',
            :rc);
```

Sample output from generating SQL

When you use the GEN command, Db2 Admin Tool generates DDL statements for the specified object or objects.

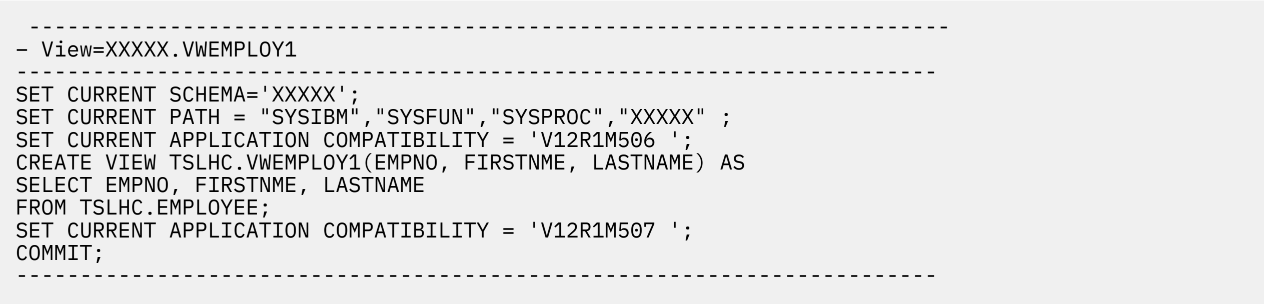
The following figures show examples of the generated SQL:

```
------------------------------------------------------------------------
--                                                                    --
-- Database 2 Administration Tool (DB2 Admin) , program 5655-W34      --
--                                                                    --
-- ADB2GEN  -  Extract object definitions from the DB2 Catalog tables --
--                                                                    --
-- Input prepared on : DSNA (1015 )   Extract time : 2013-05-14 07:37 --
--                                                                    --
-- Catalog values overridden : none
--                                                                    --
-- Generate : SG=N DB=Y TS=Y TB=Y VW=Y IX=Y SY=Y AL=Y LB=N CM=N FK=N  --
--            TG=Y UT=N UF=N SP=N SQ=N RO=N TC=N MK=Y PM=Y AC=Y       --
-- Grants   : SG=N DB=N TS=N TB=N VW=N SC=N UT=N UF=N SP=N SQ=N       --
--                                                                    --
------------------------------------------------------------------------
------------------------------------------------------------------------
--                                                                    --
-- ADB2GEN: Generate DDL for Database DSNDB06                         --
--                                                                    --
------------------------------------------------------------------------
--
--
------------------------------------------------------------------------
------------------------------------------------------------------------
--
  SET CURRENT SQLID='SYSIBM';
--
--#SET ACCEPT_RC 0 -618
--
  CREATE DATABASE DSNDB06
    INDEXBP    BP0
    CCSID      EBCDIC;
--
```

Figure 183. Sample output from generating SQL for a database

```
------------------------------------------------------------------------
-- Table space=DSNDB06.SYSALTER
------------------------------------------------------------------------
--
  CREATE TABLESPACE SYSALTER
    IN DSNDB06
--  VCAT "00000001" -- DB2 catalog tablespace
    FREEPAGE 0 PCTFREE 7
    GBPCACHE CHANGED
    TRACKMOD YES
    LOGGED
    SEGSIZE 4
    BUFFERPOOL BP32K
    LOCKSIZE ROW
    LOCKMAX SYSTEM
    CLOSE NO
    COMPRESS NO
    CCSID      UNICODE
    DEFINE YES
    MAXROWS 255;
--
```

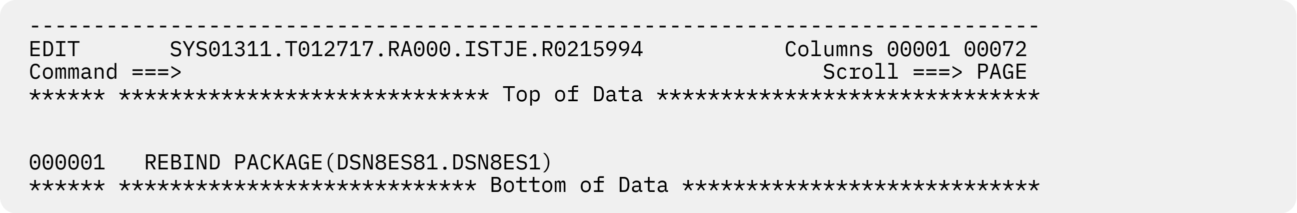
Figure 184. Sample output from generating SQL for a table space



```
 ------------------------------------------------------------------------
- View=XXXXX.VWEMPLOY1
------------------------------------------------------------------------
SET CURRENT SCHEMA='XXXXX';
SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","XXXXX" ;
SET CURRENT APPLICATION COMPATIBILITY = 'V12R1M506 ';
CREATE VIEW TSLHC.VWEMPLOY1(EMPNO, FIRSTNME, LASTNAME) AS
SELECT EMPNO, FIRSTNME, LASTNAME
FROM TSLHC.EMPLOYEE;
SET CURRENT APPLICATION COMPATIBILITY = 'V12R1M507 ';
COMMIT;
------------------------------------------------------------------------
```

Figure 185. Sample output from generating SQL for a view

If you specified that REBIND commands were to be generated, the REBIND output is also included, as shown in the following figure:



```
-------------------------------------------------------------------------------
EDIT       SYS01311.T012717.RA000.ISTJE.R0215994           Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
****** ***************************** Top of Data ******************************


000001   REBIND PACKAGE(DSN8ES81.DSN8ES1)
****** **************************** Bottom of Data ****************************
```

Figure 186. Sample output of generating SQL with the REBIND option specified

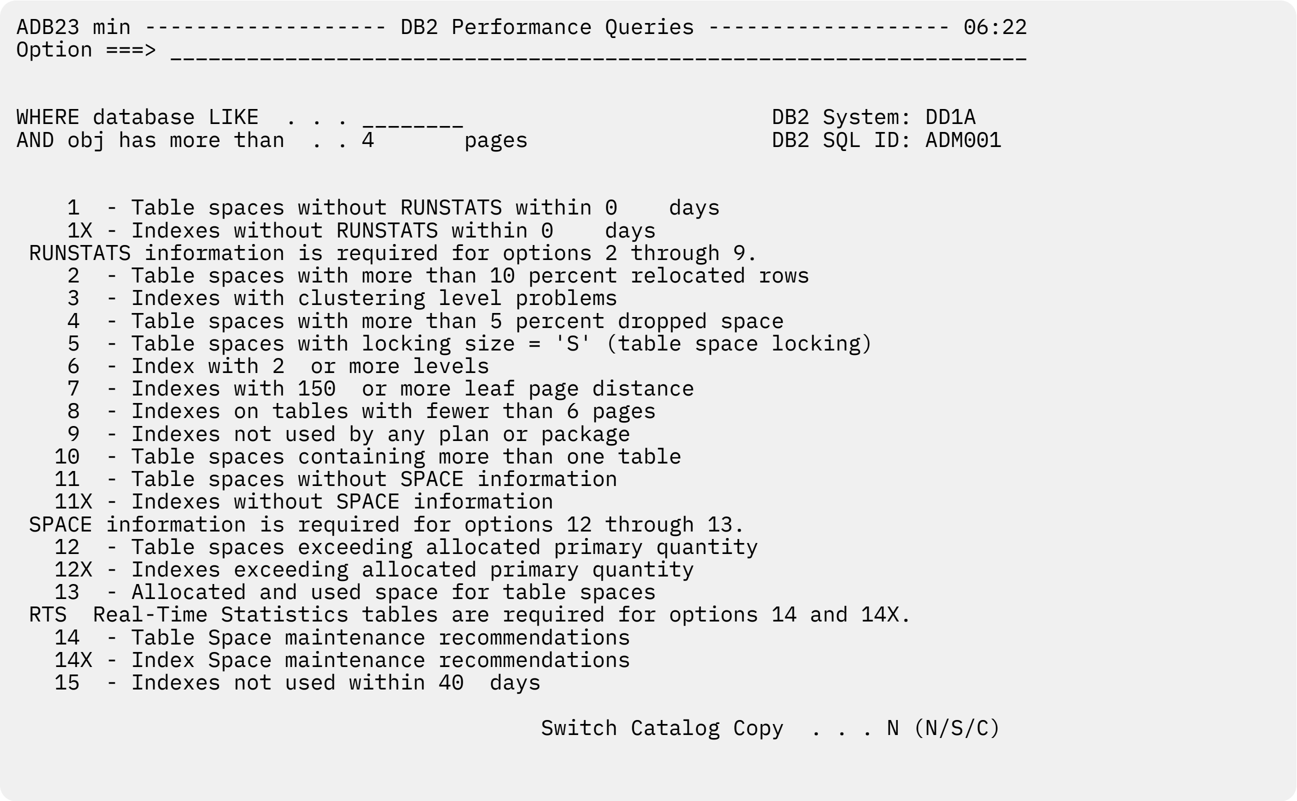
Note: In some cases, data-partitioned secondary indexes might be included in the output, because the process to generate the SQL supports these indexes.

Performance queries

This information shows you how to run performance queries by using Db2 Admin Tool and describes the different types of performance queries that Db2 Admin Tool supports.

The **DB2 Performance Queries** panel (ADB23) is displayed when you select option 3 on the **Administration Menu** panel. Use this panel to select the Db2 performance and space utilization query you want to run. Select an option, and enter (part of) the name of the database for which the query should be run. See the descriptions that appear on each panel in this chapter for more information about each option shown in the following figure.

The select field on the performance queries panels lets you select an object, which is then shown on the corresponding system catalog panel. This lets you further investigate problems or choose to run utilities such as REORG and RUNSTATS.



```
ADB23 min ------------------ DB2 Performance Queries ------------------ 06:22
Option ===> ___________________________________________________________________


WHERE database LIKE  . . . ________                        DB2 System: DD1A
AND obj has more than  . . 4       pages                   DB2 SQL ID: ADM001


    1  - Table spaces without RUNSTATS within 0    days
    1X - Indexes without RUNSTATS within 0    days
 RUNSTATS information is required for options 2 through 9.
    2  - Table spaces with more than 10 percent relocated rows
    3  - Indexes with clustering level problems
    4  - Table spaces with more than 5 percent dropped space
    5  - Table spaces with locking size = 'S' (table space locking)
    6  - Index with 2  or more levels
    7  - Indexes with 150  or more leaf page distance
    8  - Indexes on tables with fewer than 6 pages
    9  - Indexes not used by any plan or package
   10  - Table spaces containing more than one table
   11  - Table spaces without SPACE information
   11X - Indexes without SPACE information
 SPACE information is required for options 12 through 13.
   12  - Table spaces exceeding allocated primary quantity
   12X - Indexes exceeding allocated primary quantity
   13  - Allocated and used space for table spaces
 RTS  Real-Time Statistics tables are required for options 14 and 14X.
   14  - Table Space maintenance recommendations
   14X - Index Space maintenance recommendations
   15  - Indexes not used within 40  days

                                         Switch Catalog Copy  . . . N (N/S/C)
```

Figure 187. DB2 Performance Queries panel (ADB23)

Running queries on table spaces without RUNSTATS

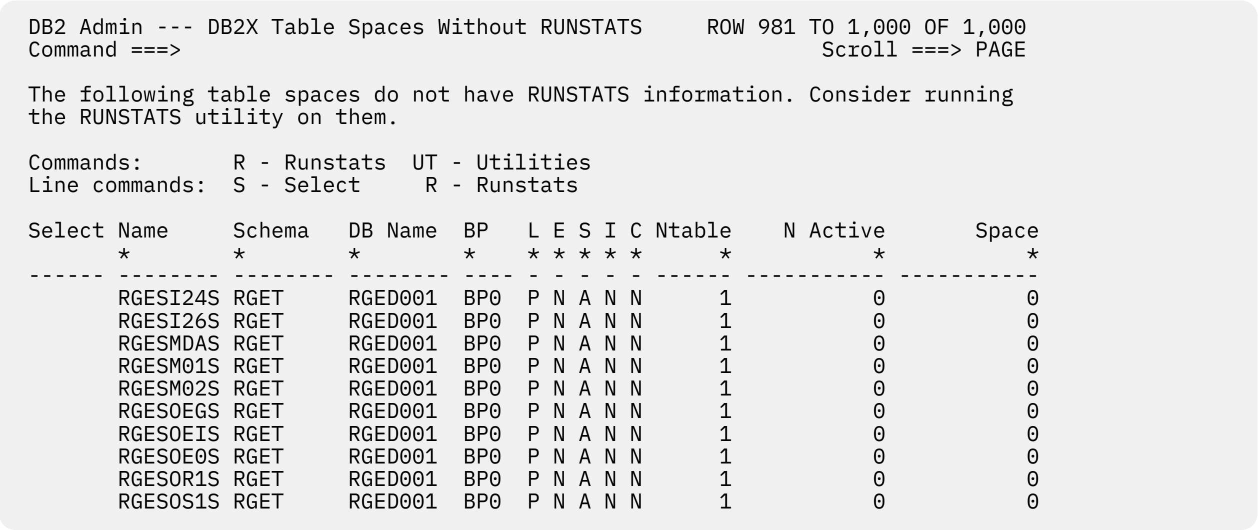
Use the performance queries function to view information about and work with table spaces that do not have RUNSTATS information.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Locate option **1 - Table spaces without RUNSTATS within n days**.
3. If necessary, change the value of *n*.
4. Specify option 1, and press Enter.

   The **Table Spaces Without RUNSTATS Information (ADB231)** panel is displayed, as shown in the following figure:



```
DB2 Admin --- DB2X Table Spaces Without RUNSTATS     ROW 981 TO 1,000 OF 1,000
Command ===>                                                  Scroll ===> PAGE

The following table spaces do not have RUNSTATS information. Consider running
the RUNSTATS utility on them.

Commands:       R - Runstats  UT - Utilities
Line commands:  S - Select     R - Runstats

Select Name     Schema   DB Name  BP   L E S I C Ntable    N Active       Space
       *        *        *        *    * * * * *      *           *           *
------ -------- -------- -------- ---- - - - - - ------ ----------- -----------
       RGESI24S RGET     RGED001  BP0  P N A N N      1           0           0
       RGESI26S RGET     RGED001  BP0  P N A N N      1           0           0
       RGESMDAS RGET     RGED001  BP0  P N A N N      1           0           0
       RGESM01S RGET     RGED001  BP0  P N A N N      1           0           0
       RGESM02S RGET     RGED001  BP0  P N A N N      1           0           0
       RGESOEGS RGET     RGED001  BP0  P N A N N      1           0           0
       RGESOEIS RGET     RGED001  BP0  P N A N N      1           0           0
       RGESOE0S RGET     RGED001  BP0  P N A N N      1           0           0
       RGESOR1S RGET     RGED001  BP0  P N A N N      1           0           0
       RGESOS1S RGET     RGED001  BP0  P N A N N      1           0           0
```

Figure 188. ***Table Spaces Without RUNSTATS Information (ADB231)*** *panel*

The following columns are on this panel:

Select
Selects a table space when you use the S line command.

Name
The name of the table space.

Schema
The name of the schema.

DB name
The name of the database.

BP
The name of the buffer pool used for the table space.

L
The locking size, which is one of the following values:

A
Any.

P
Page.

S
Table space.

E
The erase rule, which is one of the following values:

Y
Erase.

N
No erase.

S
The status of the table space, which is one of the following values:

A
Available.

N
Not available.

I

The indicator of whether the table space was created implicitly, which is one of the following values:

Y

Yes.

N

No.

C

The close rule, which is one of the following values:

Y

Yes.

N

No.

NTable

The number of tables defined in the table space.

N Active

The number of active pages in the table space. If the RUNSTATS utility was not run, the value is 0.

Space

Number of kilobytes (KB) of storage that is allocated to the table space. If the STOSPACE utility was not run, the value is 0.

5. Optional: Work with the table spaces by using the primary commands and line commands.

   Tip: For table spaces that do not have RUNSTATS information, run the RUNSTATS utility on them.

   The R line command enables you to quickly move to the **Batch Job Utility Parameters (ADB2UPA)** panel. Entering the R line command is equivalent to entering S, UT, and R commands in succession.

Running queries on indexes without RUNSTATS

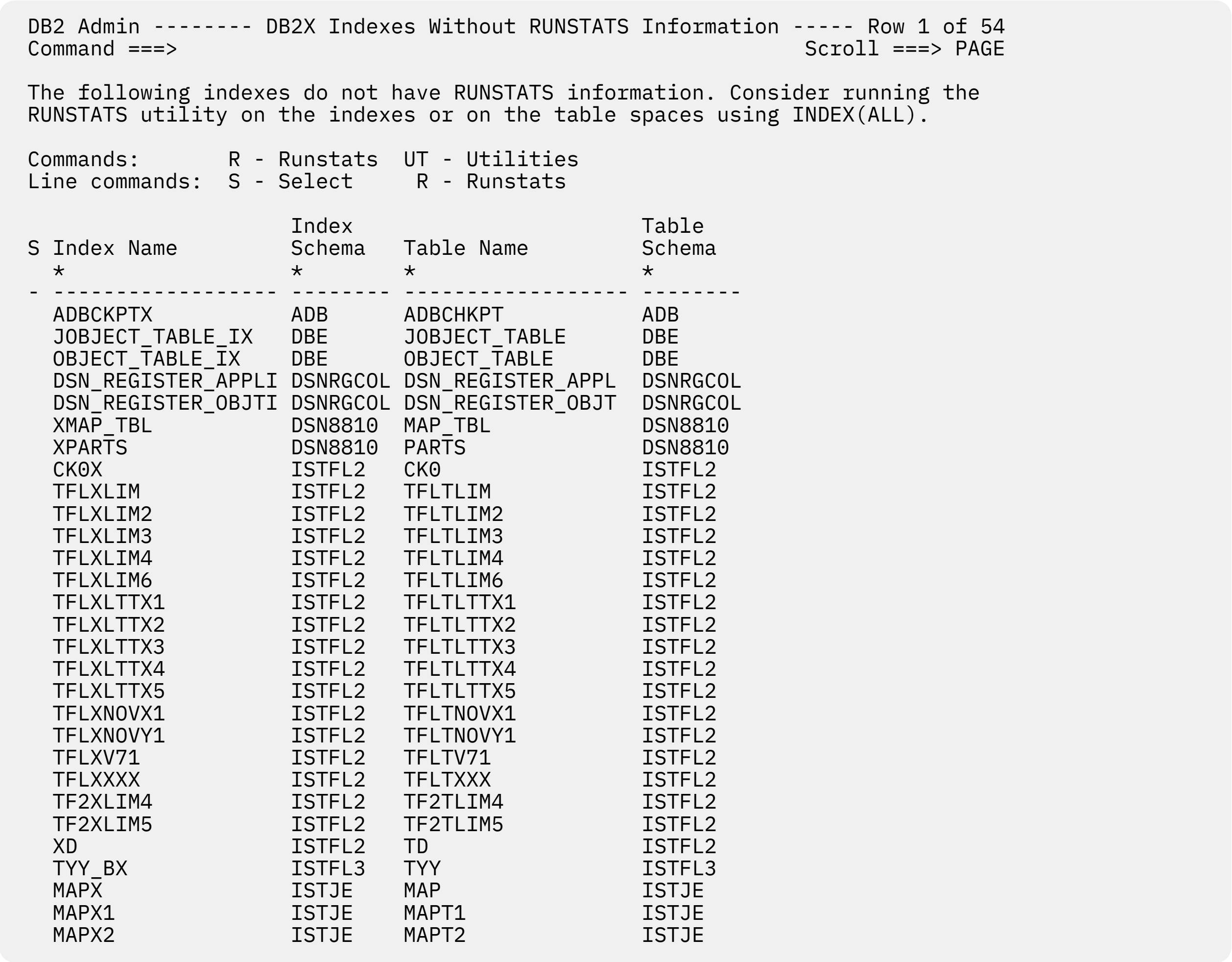
Use the performance queries function to view information about and work with indexes that do not have RUNSTATS information.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Locate option **1X - Indexes without RUNSTATS within n days**.
3. If necessary, change the value of *n*.
4. Specify option 1X, and press Enter.

   The **Indexes Without RUNSTATS Information (ADB231X)** panel is displayed, as shown in the following figure:



```
DB2 Admin -------- DB2X Indexes Without RUNSTATS Information ----- Row 1 of 54
Command ===>                                                  Scroll ===> PAGE

The following indexes do not have RUNSTATS information. Consider running the
RUNSTATS utility on the indexes or on the table spaces using INDEX(ALL).

Commands:       R - Runstats  UT - Utilities
Line commands:  S - Select     R - Runstats

                     Index                      Table
S Index Name         Schema   Table Name         Schema
  *                  *        *                  *
- ------------------ -------- ------------------ --------
  ADBCKPTX           ADB      ADBCHKPT           ADB
  JOBJECT_TABLE_IX   DBE      JOBJECT_TABLE      DBE
  OBJECT_TABLE_IX    DBE      OBJECT_TABLE       DBE
  DSN_REGISTER_APPLI DSNRGCOL DSN_REGISTER_APPL  DSNRGCOL
  DSN_REGISTER_OBJTI DSNRGCOL DSN_REGISTER_OBJT  DSNRGCOL
  XMAP_TBL           DSN8810  MAP_TBL            DSN8810
  XPARTS             DSN8810  PARTS              DSN8810
  CK0X               ISTFL2   CK0                ISTFL2
  TFLXLIM            ISTFL2   TFLTLIM            ISTFL2
  TFLXLIM2           ISTFL2   TFLTLIM2           ISTFL2
  TFLXLIM3           ISTFL2   TFLTLIM3           ISTFL2
  TFLXLIM4           ISTFL2   TFLTLIM4           ISTFL2
  TFLXLIM6           ISTFL2   TFLTLIM6           ISTFL2
  TFLXLTTX1          ISTFL2   TFLTLTTX1          ISTFL2
  TFLXLTTX2          ISTFL2   TFLTLTTX2          ISTFL2
  TFLXLTTX3          ISTFL2   TFLTLTTX3          ISTFL2
  TFLXLTTX4          ISTFL2   TFLTLTTX4          ISTFL2
  TFLXLTTX5          ISTFL2   TFLTLTTX5          ISTFL2
  TFLXNOVX1          ISTFL2   TFLTNOVX1          ISTFL2
  TFLXNOVY1          ISTFL2   TFLTNOVY1          ISTFL2
  TFLXV71            ISTFL2   TFLTV71            ISTFL2
  TFLXXXX            ISTFL2   TFLTXXX            ISTFL2
  TF2XLIM4           ISTFL2   TF2TLIM4           ISTFL2
  TF2XLIM5           ISTFL2   TF2TLIM5           ISTFL2
  XD                 ISTFL2   TD                 ISTFL2
  TYY_BX             ISTFL3   TYY                ISTFL3
  MAPX               ISTJE    MAP                ISTJE
  MAPX1              ISTJE    MAPT1              ISTJE
  MAPX2              ISTJE    MAPT2              ISTJE
```

Figure 189. ***Indexes Without RUNSTATS Information (ADB231X)*** *panel*

The following fields are shown on this panel:

S
Selects an index when you use the S line command.

Index Name
The name of the index.

Index Schema
The authorization ID of the schema of the index.

Table Name
The name of the table on which the index is defined.

Table Schema
The authorization ID of the schema of the table.

5. Optional: Work with the indexes by using the primary commands and line commands.

   Tip: For indexes that do not have RUNSTATS information, run the RUNSTATS utility on them.

   The R line command enables you to quickly move to the **Batch Job Utility Parameters (ADB2UPA)** panel. Entering the R line command is equivalent to entering S, UT, and R commands in succession.

Running queries on table spaces with more than *n* percent relocated rows

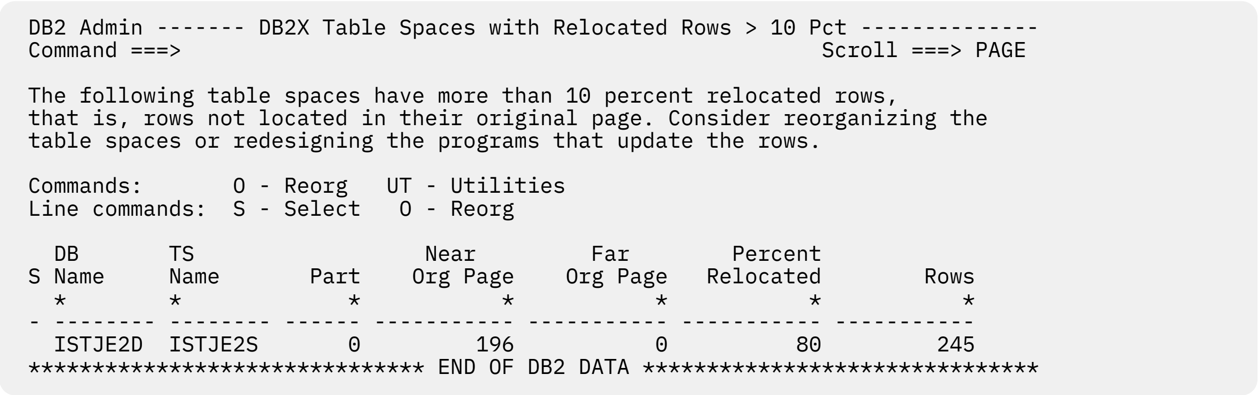
Use the performance queries function to view information about and work with tables with no more than *n* percent relocated rows.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Locate option **2 - Table spaces with more than n percent relocated rows**.
3. If necessary, change the value of *n*.
4. Specify option 2, and press Enter.

   The **Table Spaces With More Than n Percent Relocated Rows (ADB232)** panel is displayed, as shown in the following figure:



```
DB2 Admin ------- DB2X Table Spaces with Relocated Rows > 10 Pct --------------
Command ===>                                                  Scroll ===> PAGE

The following table spaces have more than 10 percent relocated rows,
that is, rows not located in their original page. Consider reorganizing the
table spaces or redesigning the programs that update the rows.

Commands:       O - Reorg   UT - Utilities
Line commands:  S - Select   O - Reorg

  DB       TS                  Near         Far        Percent
S Name     Name       Part    Org Page    Org Page   Relocated        Rows
  *        *             *           *           *           *           *
- -------- -------- ------ ----------- ----------- ----------- -----------
  ISTJE2D  ISTJE2S       0         196           0          80         245
******************************* END OF DB2 DATA *******************************
```

   Figure 190. ***Table Spaces With More Than n Percent Relocated Rows (ADB232)*** *panel*

   The following columns are on this panel:

   S
      Selects a table space when you use the S line command.

   DB Name
      The name of the database.

   TS Name
      The name of the table space.

   Part
      The partition number. If the database is not partitioned, the value is 0.

   Near Org Page
      The number of rows that have been relocated near their original page.

   Far Org Page
      The number of rows that have been relocated far from their original page.

   Percent Relocated
      The percentage of rows that have been relocated.

   Rows
      The number of rows in the table space or partition.
5. Optional: Work with the table spaces by using the primary commands and line commands.

   Tip: For table spaces that have more than 10 percent relocated rows, that is, rows that are not located in their original page, reorganize the table spaces or review the pctfree and/or the free page values to leave more space for rows to grow during an update.

   The O line command enables you to quickly move to the **Batch Job Utility Parameters (ADB2UPA)** panel. Entering the O line command is equivalent to entering S, UT, and O commands in succession.

Running queries on indexes with clustering-level problems

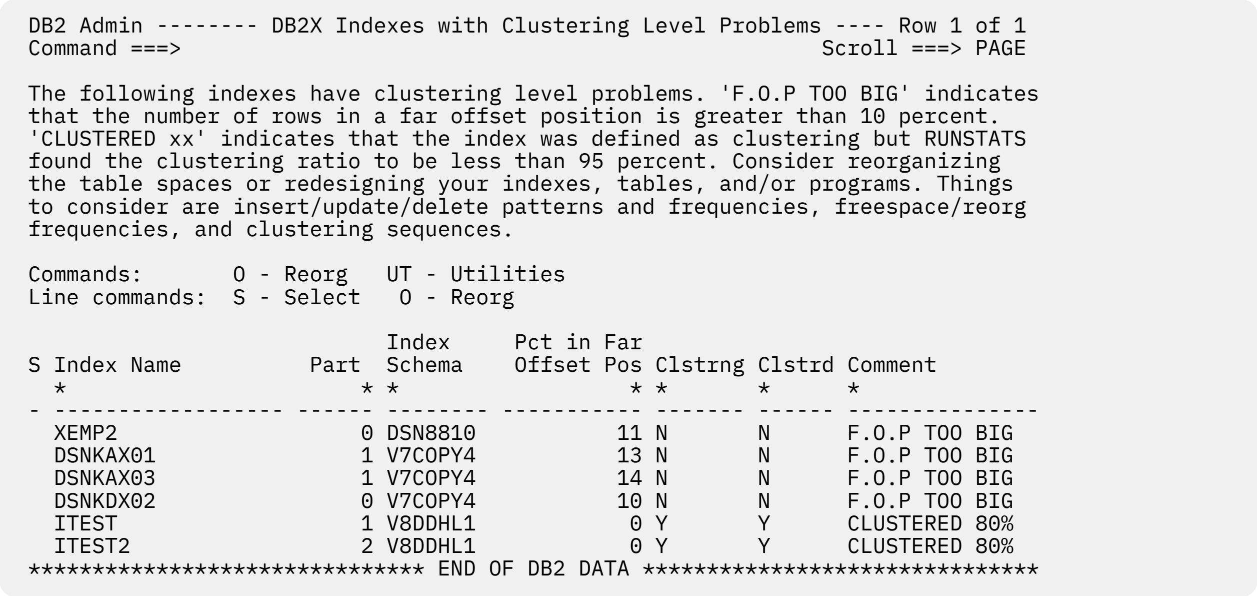
Use the performance queries function to view information about and work with indexes that have clustering-level problems.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Specify option 3, and press Enter.

   The **Indexes with Clustering Level Problems (ADB233)** panel is displayed, as shown in the following figure:



```
DB2 Admin -------- DB2X Indexes with Clustering Level Problems ---- Row 1 of 1
Command ===>                                                  Scroll ===> PAGE

The following indexes have clustering level problems. 'F.O.P TOO BIG' indicates
that the number of rows in a far offset position is greater than 10 percent.
'CLUSTERED xx' indicates that the index was defined as clustering but RUNSTATS
found the clustering ratio to be less than 95 percent. Consider reorganizing
the table spaces or redesigning your indexes, tables, and/or programs. Things
to consider are insert/update/delete patterns and frequencies, freespace/reorg
frequencies, and clustering sequences.

Commands:       O - Reorg   UT - Utilities
Line commands:  S - Select   O - Reorg

                            Index     Pct in Far
S Index Name          Part  Schema    Offset Pos Clstrng Clstrd Comment
  *                       * *                  * *       *      *
- ------------------ ------ -------- ----------- ------- ------ ---------------
  XEMP2                   0 DSN8810           11 N       N      F.O.P TOO BIG
  DSNKAX01                1 V7COPY4           13 N       N      F.O.P TOO BIG
  DSNKAX03                1 V7COPY4           14 N       N      F.O.P TOO BIG
  DSNKDX02                0 V7COPY4           10 N       N      F.O.P TOO BIG
  ITEST                   1 V8DDHL1            0 Y       Y      CLUSTERED 80%
  ITEST2                  2 V8DDHL1            0 Y       Y      CLUSTERED 80%
******************************* END OF DB2 DATA *******************************
```

*Figure 191. **Indexes with Clustering Level Problems (ADB233)** panel*

The following columns are on this panel:

S
: Selects an index when you use the S line command.

Index Name
: The name of the index.

Part
: The partition number. If the database is not partitioned, the value is 0.

Index Schema
: The authorization ID of the schema of the index.

Pct in Far Offset Pos
: Percentage in Far Offset Position. This column shows percentage of rows in a far offset position because of an insert into a full page.

Clstrng
: Clustering. This column shows whether CLUSTER was specified when the index was created.

Clstrd
: Clustered. This column shows whether the table is actually clustered by the index.

Comment
: Comments about the reason why the index appears in the list.

For indexes that have clustering level problems, the message `F.O.P TOO BIG` is displayed and indicates that the number of rows in a far offset position is greater than 10 percent. Additionally, `CLUSTERED` *nn%* indicates that the index was defined as clustering, but the RUNSTATS utility found the clustering ratio to be less than 95 percent.

Consider reorganizing the table spaces or redesigning your indexes, tables, and programs. Consider the insert/update/delete patterns and frequencies, freespace/reorganization frequencies, and clustering sequences.

3. Optional: Work with the indexes by using the primary commands and line commands.

   The O line command enables you to quickly move to the **Batch Job Utility Parameters (ADB2UPA)** panel. Entering the O line command is equivalent to entering S, UT, and O commands in succession.

Running queries on table spaces with more than *n* percent dropped space

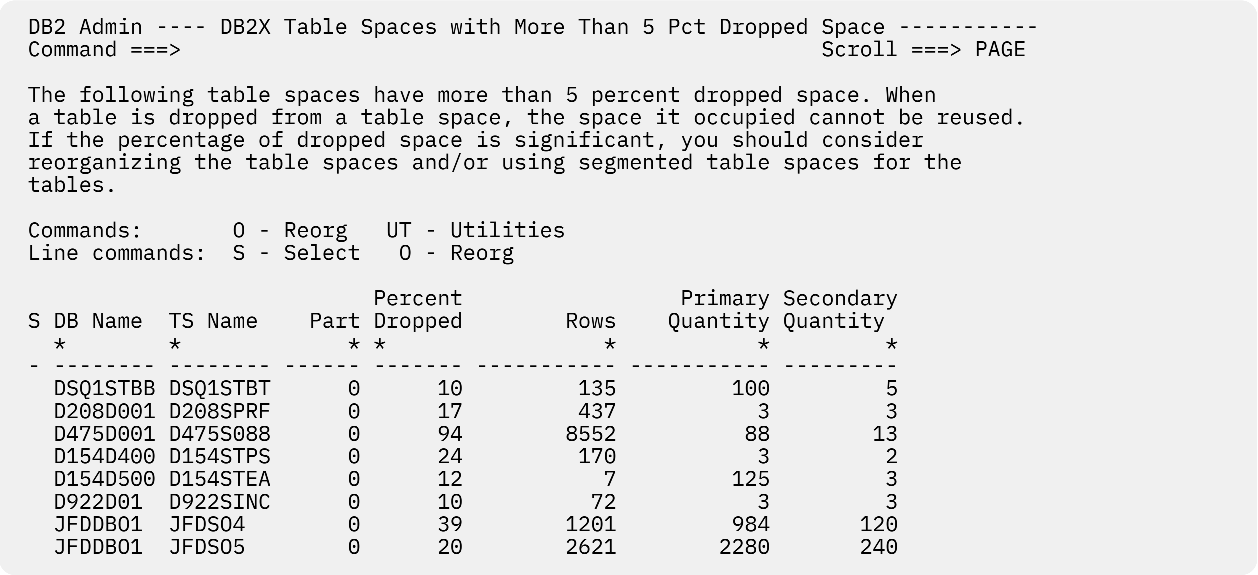
Use the performance queries function to view information about and work with table spaces that have more than *n* percent of dropped space.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Locate option **4 - Table Spaces with More Than *n* Percent Dropped Space**.
3. If necessary, change the value of *n*.
4. Specify option 4, and press Enter.

   The **Table Spaces With More Than n Percent Dropped Space (ADB234)** panel is displayed, as shown in the following figure:



```
DB2 Admin ---- DB2X Table Spaces with More Than 5 Pct Dropped Space -----------
Command ===>                                                  Scroll ===> PAGE

The following table spaces have more than 5 percent dropped space. When
a table is dropped from a table space, the space it occupied cannot be reused.
If the percentage of dropped space is significant, you should consider
reorganizing the table spaces and/or using segmented table spaces for the
tables.

Commands:       O - Reorg   UT - Utilities
Line commands:  S - Select   O - Reorg

                           Percent                 Primary Secondary
S DB Name  TS Name    Part Dropped        Rows    Quantity Quantity
  *        *             * *                 *           *         *
- -------- -------- ------ ------- ----------- ----------- ---------
  DSQ1STBB DSQ1STBT      0      10         135         100         5
  D208D001 D208SPRF      0      17         437           3         3
  D475D001 D475S088      0      94        8552          88        13
  D154D400 D154STPS      0      24         170           3         2
  D154D500 D154STEA      0      12           7         125         3
  D922D01  D922SINC      0      10          72           3         3
  JFDDBO1  JFDSO4        0      39        1201         984       120
  JFDDBO1  JFDSO5        0      20        2621        2280       240
```

*Figure 192. **Table Spaces With More Than n Percent Dropped Space (ADB234)** panel*

The following columns are on this panel:

S
  Selects a table space when you use the S line command.

DB Name
  The name of the database.

TS Name
  The name of the table space.

Part
  The partition number. If the database is not partitioned, the value is 0.

Percent Dropped
The percentage of space that is occupied by dropped tables.

Rows
The number of rows in the table space or partition.

Primary Quantity
The primary space allocation in 4K blocks of storage.

Secondary Quantity
The secondary space allocation in 4K blocks of storage.

5. Optional: Work with the table spaces by using the primary commands and line commands.

   When a table is dropped from a table space, the space it occupied cannot be reused. If the percent of dropped space is significant, consider reorganizing the table spaces and use segmented table spaces for the tables.

   Tip:

   If some of your table spaces have dropped tables, consider running the MODIFY utility to remove the details of the table from the DBD.

   The O line command enables you to quickly move to the **Batch Job Utility Parameters (ADB2UPA)** panel. Entering the O line command is equivalent to entering S, UT, and O commands in succession.

Running queries on Db2 table spaces with locking size = 'S'

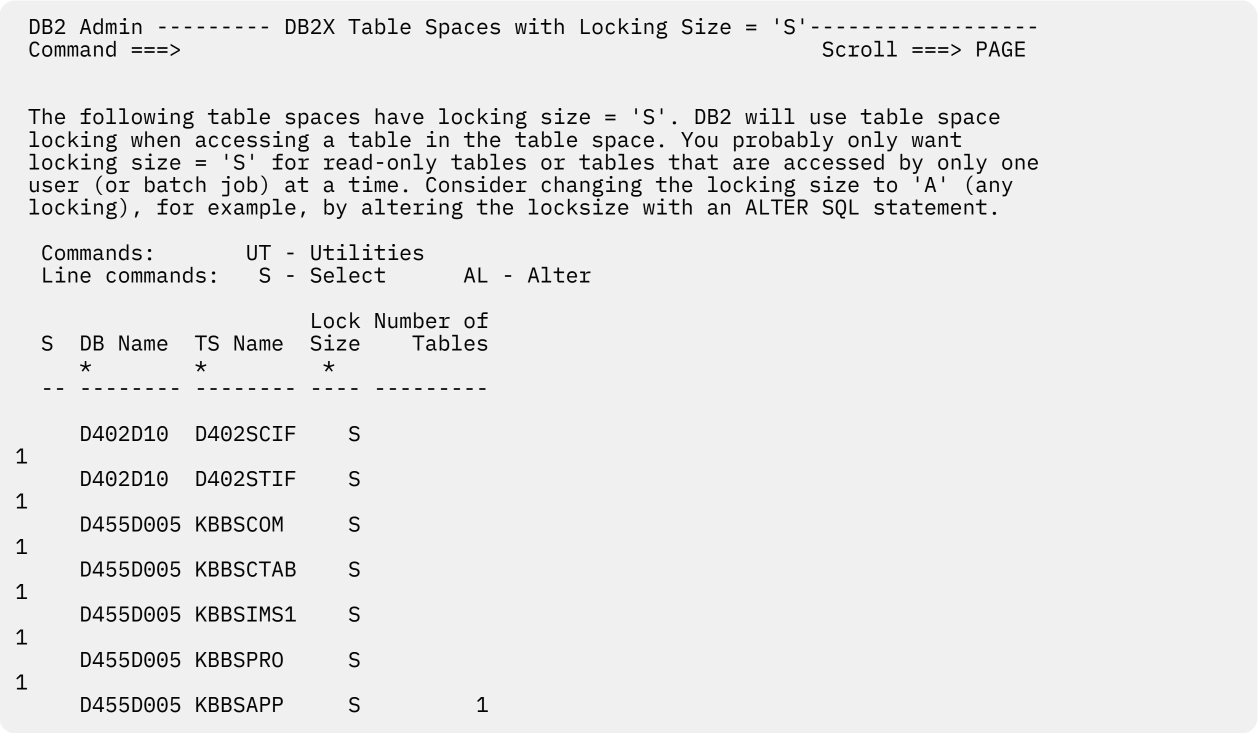
Use the performance queries function to view information about and work with table spaces with locking size = 'S'.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Specify option 5, and press Enter.

   The **DB2 Table Spaces With Locking Size = 'S' (ADB235)** panel panel is displayed, as shown in the following figure:



```
 DB2 Admin --------- DB2X Table Spaces with Locking Size = 'S'------------------
 Command ===>                                                  Scroll ===> PAGE


 The following table spaces have locking size = 'S'. DB2 will use table space
 locking when accessing a table in the table space. You probably only want
 locking size = 'S' for read-only tables or tables that are accessed by only one
 user (or batch job) at a time. Consider changing the locking size to 'A' (any
 locking), for example, by altering the locksize with an ALTER SQL statement.

  Commands:       UT - Utilities
  Line commands:   S - Select      AL - Alter

                       Lock Number of
  S  DB Name  TS Name  Size    Tables
     *        *         *
  -- -------- -------- ---- ---------

     D402D10  D402SCIF    S
1
     D402D10  D402STIF    S
1
     D455D005 KBBSCOM     S
1
     D455D005 KBBSCTAB    S
1
     D455D005 KBBSIMS1    S
1
     D455D005 KBBSPRO     S
1
     D455D005 KBBSAPP     S         1
```

*Figure 193. **DB2 Table Spaces With Locking Size = 'S' (ADB235)** panel*

The following columns are on this panel:

S
: Selects a table space when you use the S line command.

DB Name
: The name of the database.

TS Name
: The name of the table space.

Lock Size
: The lock size of the table space.

Number of Tables
: The number of tables defined in the table space.

3. Optional: Work with the table spaces by using the primary commands and line commands.

   Db2 uses table space locking when it accesses a table in the table space. Only use locking size = 'S' for read-only tables or tables that are accessed by only one user (or batch job) at a time. If concurrency between updating tasks or updaters and readers is required, then consider changing the locking size to 'A' (any locking) by altering the locksize with an ALTER SQL statement.

   The AL line command enables you to quickly perform an ALTER TABLESPACE statement to change the LOCK SIZE to ANY. Entering the AL line command is equivalent to entering the S line command followed by the AL line command, and then entering ANY in the **LOCK SIZE** field.

   The O line command enables you to quickly move to the **Batch Job Utility Parameters (ADB2UPA)** panel. Entering the O line command is equivalent to entering S, UT, and O commands in succession.

Running queries on indexes with *n* or more levels

Use the performance queries function to view information about and work with indexes with *n* or more levels.

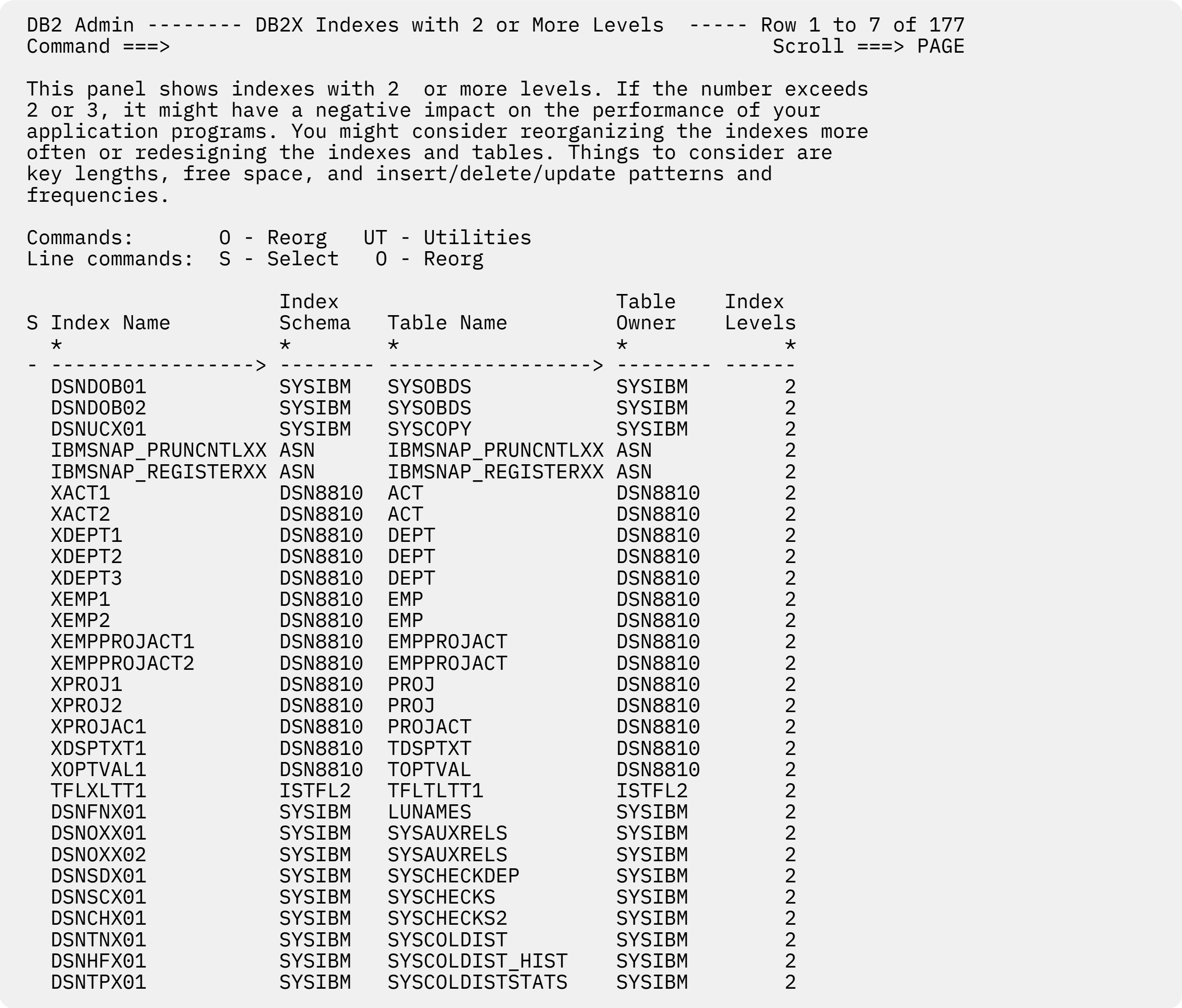
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Locate option **6 - Index with *n* or more levels**.
3. If necessary, change the value of *n*.

   Valid values are 2 - 99.
4. Specify option 6, and press Enter.

   The **Indexes with n or More Levels (ADB236)** panel is displayed, as shown in the following figure:



```
DB2 Admin -------- DB2X Indexes with 2 or More Levels  ----- Row 1 to 7 of 177
Command ===>                                                  Scroll ===> PAGE

This panel shows indexes with 2  or more levels. If the number exceeds
2 or 3, it might have a negative impact on the performance of your
application programs. You might consider reorganizing the indexes more
often or redesigning the indexes and tables. Things to consider are
key lengths, free space, and insert/delete/update patterns and
frequencies.

Commands:       O - Reorg   UT - Utilities
Line commands:  S - Select   O - Reorg

                     Index                         Table    Index
S Index Name         Schema   Table Name         Owner    Levels
  *                  *        *                  *             *
- -----------------> -------- -----------------> -------- ------
  DSNDOB01           SYSIBM   SYSOBDS            SYSIBM        2
  DSNDOB02           SYSIBM   SYSOBDS            SYSIBM        2
  DSNUCX01           SYSIBM   SYSCOPY            SYSIBM        2
  IBMSNAP_PRUNCNTLXX ASN      IBMSNAP_PRUNCNTLXX ASN           2
  IBMSNAP_REGISTERXX ASN      IBMSNAP_REGISTERXX ASN           2
  XACT1              DSN8810  ACT                DSN8810       2
  XACT2              DSN8810  ACT                DSN8810       2
  XDEPT1             DSN8810  DEPT               DSN8810       2
  XDEPT2             DSN8810  DEPT               DSN8810       2
  XDEPT3             DSN8810  DEPT               DSN8810       2
  XEMP1              DSN8810  EMP                DSN8810       2
  XEMP2              DSN8810  EMP                DSN8810       2
  XEMPPROJACT1       DSN8810  EMPPROJACT         DSN8810       2
  XEMPPROJACT2       DSN8810  EMPPROJACT         DSN8810       2
  XPROJ1             DSN8810  PROJ               DSN8810       2
  XPROJ2             DSN8810  PROJ               DSN8810       2
  XPROJAC1           DSN8810  PROJACT            DSN8810       2
  XDSPTXT1           DSN8810  TDSPTXT            DSN8810       2
  XOPTVAL1           DSN8810  TOPTVAL            DSN8810       2
  TFLXLTT1           ISTFL2   TFLTLTT1           ISTFL2        2
  DSNFNX01           SYSIBM   LUNAMES            SYSIBM        2
  DSNOXX01           SYSIBM   SYSAUXRELS         SYSIBM        2
  DSNOXX02           SYSIBM   SYSAUXRELS         SYSIBM        2
  DSNSDX01           SYSIBM   SYSCHECKDEP        SYSIBM        2
  DSNSCX01           SYSIBM   SYSCHECKS          SYSIBM        2
  DSNCHX01           SYSIBM   SYSCHECKS2         SYSIBM        2
  DSNTNX01           SYSIBM   SYSCOLDIST         SYSIBM        2
  DSNHFX01           SYSIBM   SYSCOLDIST_HIST    SYSIBM        2
  DSNTPX01           SYSIBM   SYSCOLDISTSTATS    SYSIBM        2
```

*Figure 194. **Indexes with n or More Levels (ADB236)** panel*

The following columns are on this panel:

S

Selects an index when you use the S line command.

Index Name
The name of the index.

Index Schema
The authorization ID of the owner of the index.

Table Name
The name of the table on which the index is defined.

Table Owner
The authorization ID of the owner of the table.

Index Levels
The number of levels in the index tree.

5. Optional: Work with the indexes by using the primary commands and line commands.

   The **Indexes with n or More Levels (ADB236)** panel shows the number of index levels. If the number exceeds 2 or 3, the performance of your application programs might suffer. Consider reorganizing the indexes more often or redesigning the indexes and tables. Consider key lengths, free space (pctfree and/or freepage), and insert/delete/update patterns and frequencies.

   The O line command enables you to quickly move to the **Batch Job Utility Parameters (ADB2UPA)** panel. Entering the O line command is equivalent to entering S, UT, and O commands in succession.

Running queries on indexes with *n* or more leaf page distance

Use the performance queries function to view information about and work with indexes with *n* or more leaf page distance.

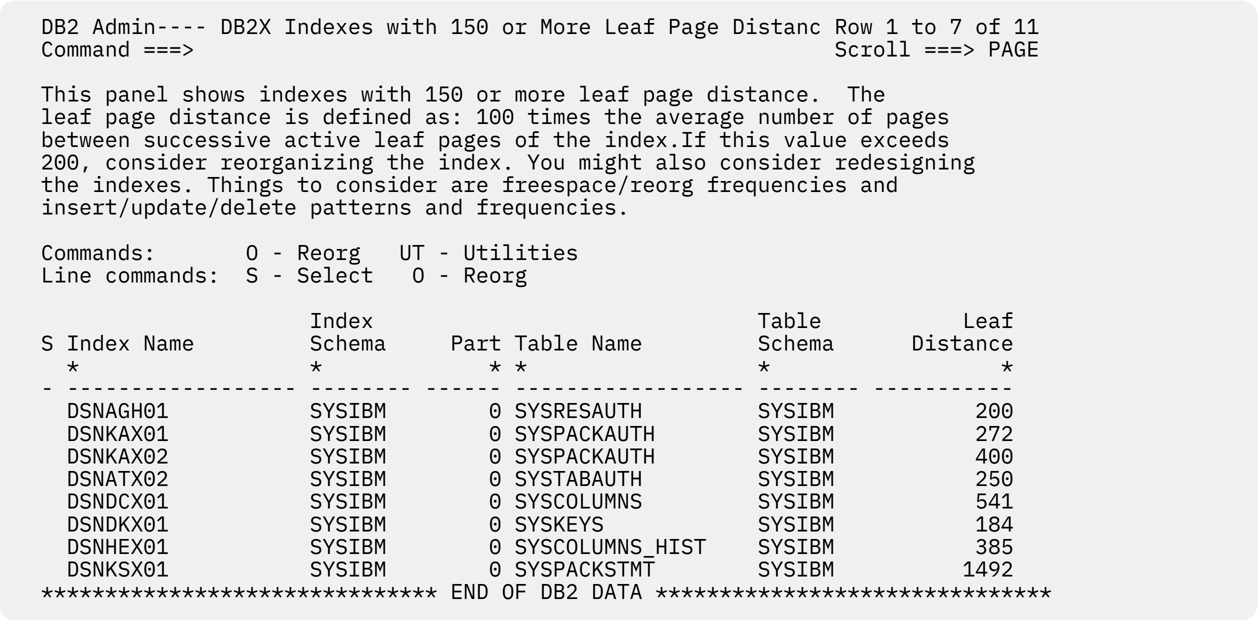
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Locate option **7 - Indexes with *n* or more leaf page distance**.
3. If necessary, change the value of *n*.

   Valid values are 150 - 9999.
4. Specify option 7, and press Enter.

   The **Indexes with 150 or more Leaf Page Distance (ADB237)** panel is displayed, as shown in the following figure:



```
DB2 Admin---- DB2X Indexes with 150 or More Leaf Page Distanc Row 1 to 7 of 11
Command ===>                                                  Scroll ===> PAGE

This panel shows indexes with 150 or more leaf page distance.  The
leaf page distance is defined as: 100 times the average number of pages
between successive active leaf pages of the index.If this value exceeds
200, consider reorganizing the index. You might also consider redesigning
the indexes. Things to consider are freespace/reorg frequencies and
insert/update/delete patterns and frequencies.

Commands:       O - Reorg   UT - Utilities
Line commands:  S - Select   O - Reorg

                     Index                             Table           Leaf
S Index Name         Schema     Part Table Name         Schema      Distance
  *                  *             * *                  *                  *
- ------------------ -------- ------ ------------------ -------- -----------
  DSNAGH01           SYSIBM        0 SYSRESAUTH         SYSIBM           200
  DSNKAX01           SYSIBM        0 SYSPACKAUTH        SYSIBM           272
  DSNKAX02           SYSIBM        0 SYSPACKAUTH        SYSIBM           400
  DSNATX02           SYSIBM        0 SYSTABAUTH         SYSIBM           250
  DSNDCX01           SYSIBM        0 SYSCOLUMNS         SYSIBM           541
  DSNDKX01           SYSIBM        0 SYSKEYS            SYSIBM           184
  DSNHEX01           SYSIBM        0 SYSCOLUMNS_HIST    SYSIBM           385
  DSNKSX01           SYSIBM        0 SYSPACKSTMT        SYSIBM          1492
******************************* END OF DB2 DATA *******************************
```

Figure 195. ***Indexes with 150 or more Leaf Page Distance (ADB237)*** *panel*

The following columns are on this panel:

S
Selects an index when you use the S line command.

Index Name
The name of the index.

Index Schema
The authorization ID of the owner of the index.

Part
The partition number. If it's not partitioned, the value is 0.

Table Name
The name of the table on which the index is defined.

Table Owner
The authorization ID of the owner of the table.

Leaf Distance
One hundred times the average number of leaf pages between successive active leaf pages of the index. If this value exceeds 200, consider reorganizing the index. Also, consider redesigning the indexes. Consider free space/reorganization frequencies and insert/update/delete patterns and frequencies.

5. Optional: Work with the indexes by using the primary commands and line commands.

   The O line command enables you to quickly move to the **Batch Job Utility Parameters (ADB2UPA)** panel. Entering the O line command is equivalent to entering S, UT, and O commands in succession.

Running queries on indexes on tables with fewer than *n* pages

Use the performance queries function to view information about and work with indexes on tables with fewer than *n* pages.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

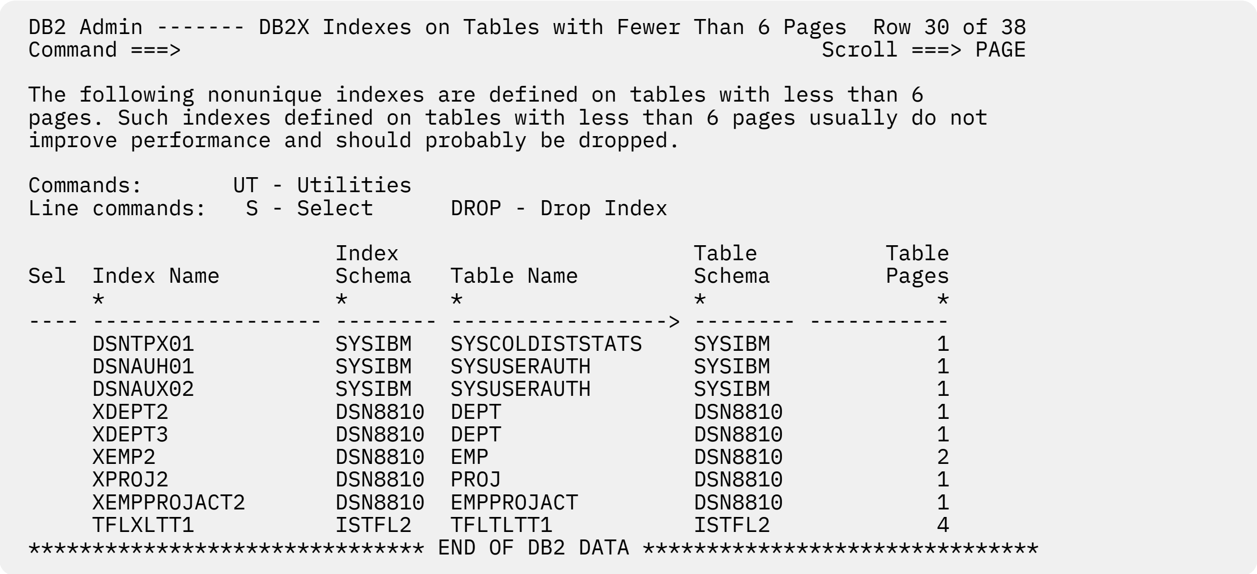
   The **DB2 Performance Queries** panel is displayed.
2. Locate option **8 - Indexes on tables with fewer than *n* pages**.

3. If necessary, change the value of *n*.

   Valid values are 150 - 9999.

4. Specify option 8, and press Enter.

   The **Indexes On Tables With Fewer Than n Pages (ADB238)** panel is displayed, as shown in the following figure:



```
DB2 Admin ------- DB2X Indexes on Tables with Fewer Than 6 Pages  Row 30 of 38
Command ===>                                                  Scroll ===> PAGE

The following nonunique indexes are defined on tables with less than 6
pages. Such indexes defined on tables with less than 6 pages usually do not
improve performance and should probably be dropped.

Commands:       UT - Utilities
Line commands:   S - Select      DROP - Drop Index

                        Index                       Table          Table
Sel  Index Name         Schema   Table Name         Schema         Pages
     *                  *        *                  *                  *
---- ------------------ -------- -----------------> -------- -----------
     DSNTPX01           SYSIBM   SYSCOLDISTSTATS    SYSIBM             1
     DSNAUH01           SYSIBM   SYSUSERAUTH        SYSIBM             1
     DSNAUX02           SYSIBM   SYSUSERAUTH        SYSIBM             1
     XDEPT2             DSN8810  DEPT               DSN8810            1
     XDEPT3             DSN8810  DEPT               DSN8810            1
     XEMP2              DSN8810  EMP                DSN8810            2
     XPROJ2             DSN8810  PROJ               DSN8810            1
     XEMPPROJACT2       DSN8810  EMPPROJACT         DSN8810            1
     TFLXLTT1           ISTFL2   TFLTLTT1           ISTFL2             4
******************************* END OF DB2 DATA *******************************
```

Figure 196. ***Indexes On Tables With Fewer Than n Pages (ADB238)*** *panel*

   The following columns are on this panel:

   Sel
      Selects an index when you use the S line command.

   Index Name
      The name of the index.

   Index Schema
      The authorization ID of the owner of the index.

   Table Name
      The name of the table on which the index is defined.

   Table Schema
      The authorization ID of the owner of the table.

   Table Pages
      The total number of pages on which rows of the table are displayed.

5. Optional: Work with the indexes by using the primary commands and line commands.

   Consider dropping nonunique indexes that are defined on tables that have less than 6 pages. Unless the index is on a table in a table space that has multiple tables, it is unlikely to improve performance but will use resources to maintain its viability. However, do not drop unique indexes, indexes supporting constraints, clustering indexes, or the only index on a table without a full evaluation.

   The DROP line command enables you to quickly issue a DROP INDEX statement. Entering the DROP line command is equivalent to entering an S line command and a DROP line command in succession.

Running queries on indexes not used by any plan or package

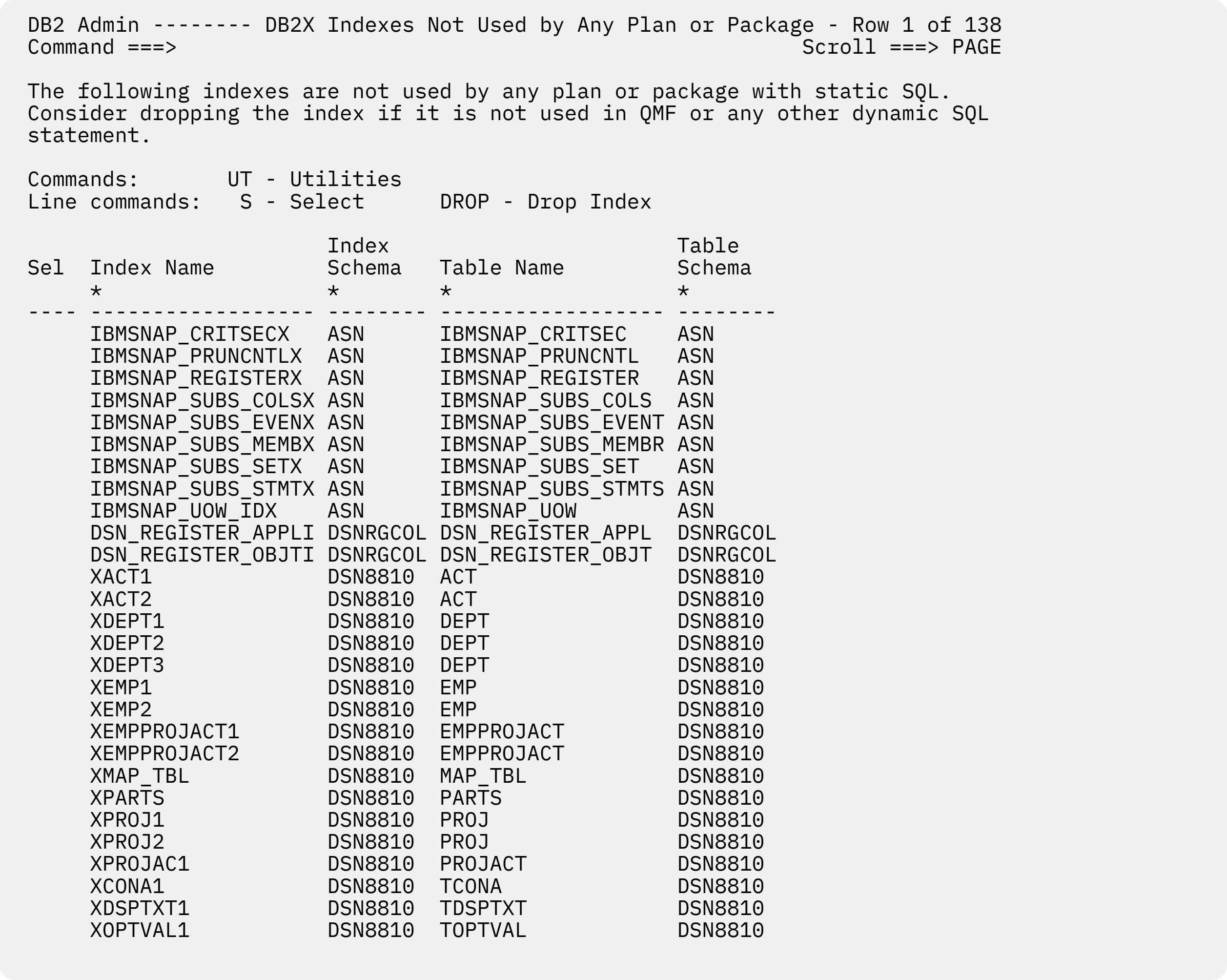
Use the performance queries function to view information about and work with indexes that are not used by any plan or package.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Specify option 8, and press Enter.

   The **Indexes Not Used By Any Plan or Package (ADB239)** panel panel is displayed, as shown in the following figure:



```
DB2 Admin -------- DB2X Indexes Not Used by Any Plan or Package - Row 1 of 138
Command ===>                                                   Scroll ===> PAGE

The following indexes are not used by any plan or package with static SQL.
Consider dropping the index if it is not used in QMF or any other dynamic SQL
statement.

Commands:       UT - Utilities
Line commands:   S - Select      DROP - Drop Index

                        Index                       Table
Sel  Index Name         Schema   Table Name         Schema
     *                  *        *                  *
---- ------------------ -------- ------------------ --------
     IBMSNAP_CRITSECX   ASN      IBMSNAP_CRITSEC    ASN
     IBMSNAP_PRUNCNTLX  ASN      IBMSNAP_PRUNCNTL   ASN
     IBMSNAP_REGISTERX  ASN      IBMSNAP_REGISTER   ASN
     IBMSNAP_SUBS_COLSX ASN      IBMSNAP_SUBS_COLS  ASN
     IBMSNAP_SUBS_EVENX ASN      IBMSNAP_SUBS_EVENT ASN
     IBMSNAP_SUBS_MEMBX ASN      IBMSNAP_SUBS_MEMBR ASN
     IBMSNAP_SUBS_SETX  ASN      IBMSNAP_SUBS_SET   ASN
     IBMSNAP_SUBS_STMTX ASN      IBMSNAP_SUBS_STMTS ASN
     IBMSNAP_UOW_IDX    ASN      IBMSNAP_UOW        ASN
     DSN_REGISTER_APPLI DSNRGCOL DSN_REGISTER_APPL  DSNRGCOL
     DSN_REGISTER_OBJTI DSNRGCOL DSN_REGISTER_OBJT  DSNRGCOL
     XACT1              DSN8810  ACT                DSN8810
     XACT2              DSN8810  ACT                DSN8810
     XDEPT1             DSN8810  DEPT               DSN8810
     XDEPT2             DSN8810  DEPT               DSN8810
     XDEPT3             DSN8810  DEPT               DSN8810
     XEMP1              DSN8810  EMP                DSN8810
     XEMP2              DSN8810  EMP                DSN8810
     XEMPPROJACT1       DSN8810  EMPPROJACT         DSN8810
     XEMPPROJACT2       DSN8810  EMPPROJACT         DSN8810
     XMAP_TBL           DSN8810  MAP_TBL            DSN8810
     XPARTS             DSN8810  PARTS              DSN8810
     XPROJ1             DSN8810  PROJ               DSN8810
     XPROJ2             DSN8810  PROJ               DSN8810
     XPROJAC1           DSN8810  PROJACT            DSN8810
     XCONA1             DSN8810  TCONA              DSN8810
     XDSPTXT1           DSN8810  TDSPTXT            DSN8810
     XOPTVAL1           DSN8810  TOPTVAL            DSN8810
```

*Figure 197. **Indexes Not Used By Any Plan or Package (ADB239)** panel*

The following columns are on this panel:

Sel
Selects an index when you use the S line command.

Index Name
The name of the index.

Index Schema
The authorization ID of the owner of the index.

Table Name
The name of the table on which the index is defined.

Table Schema
The authorization ID of the owner of the table.

3. Optional: Work with the indexes by using the primary commands and line commands.

   Consider dropping indexes that are not used by any plan or package with static SQL if they are not used in QMF or any other dynamic SQL statement.

   The DROP line command enables you to quickly issue a DROP INDEX statement. Entering the DROP line command is equivalent to entering an S line command and a DROP line command in succession.

Running queries on table spaces containing more than one table

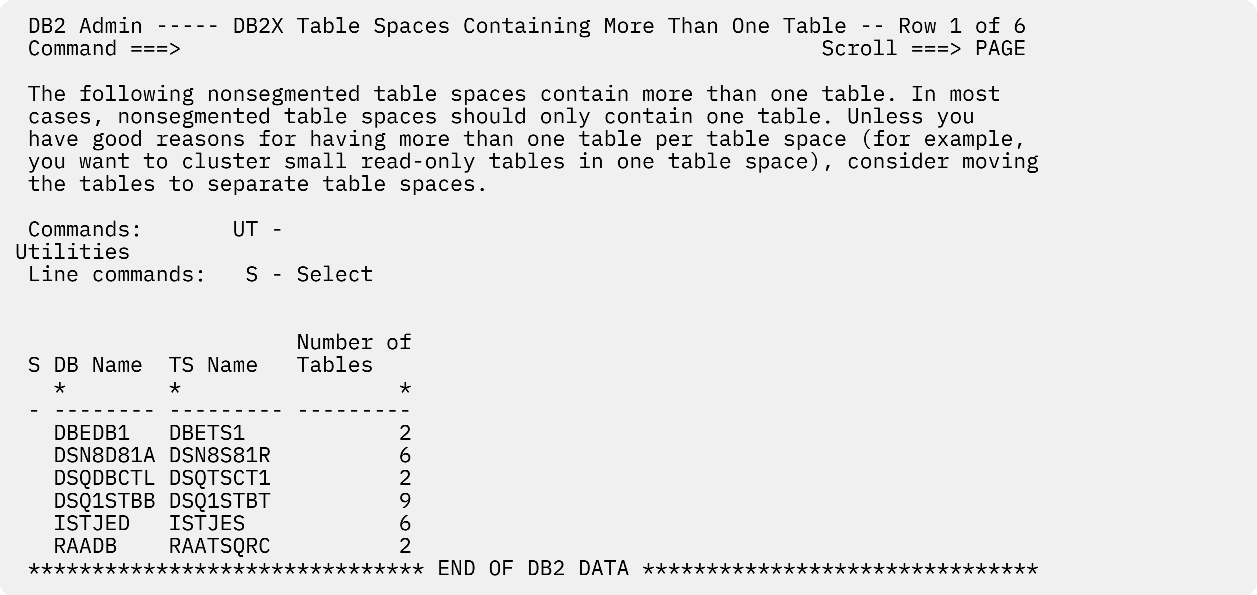
Use the performance queries function to view information about and work with table spaces containing multiple tables.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Specify option 10, and press Enter.

   The **Table Spaces Containing More Than One Table (ADB2310)** panel is displayed, as shown in the following figure:



```
 DB2 Admin ----- DB2X Table Spaces Containing More Than One Table -- Row 1 of 6
 Command ===>                                                  Scroll ===> PAGE

 The following nonsegmented table spaces contain more than one table. In most
 cases, nonsegmented table spaces should only contain one table. Unless you
 have good reasons for having more than one table per table space (for example,
 you want to cluster small read-only tables in one table space), consider moving
 the tables to separate table spaces.

 Commands:       UT -
Utilities
 Line commands:   S - Select


                      Number of
 S DB Name  TS Name   Tables
   *        *                 *
 - -------- --------- ---------
   DBEDB1   DBETS1            2
   DSN8D81A DSN8S81R          6
   DSQDBCTL DSQTSCT1          2
   DSQ1STBB DSQ1STBT          9
   ISTJED   ISTJES            6
   RAADB    RAATSQRC          2
 ****************************** END OF DB2 DATA *******************************
```

Figure 198. ***Table Spaces Containing More Than One Table (ADB2310)*** *panel*

   The following columns are on this panel:

   S
   Selects a table space when you use the S line command.

   DB Name
   The name of the database.

   Table Name
   The name of the table on which the index is defined.

   Number of tables
   The number of tables that are defined in the table space.
3. Optional: Work with the table spaces by using the primary commands and line commands.

In general, nonsegmented table spaces should only contain one table. Unless you require more than one table per table space (for example, if you want to cluster small read-only tables in one table space), consider moving the tables to separate table spaces.

Running queries on table spaces without SPACE information

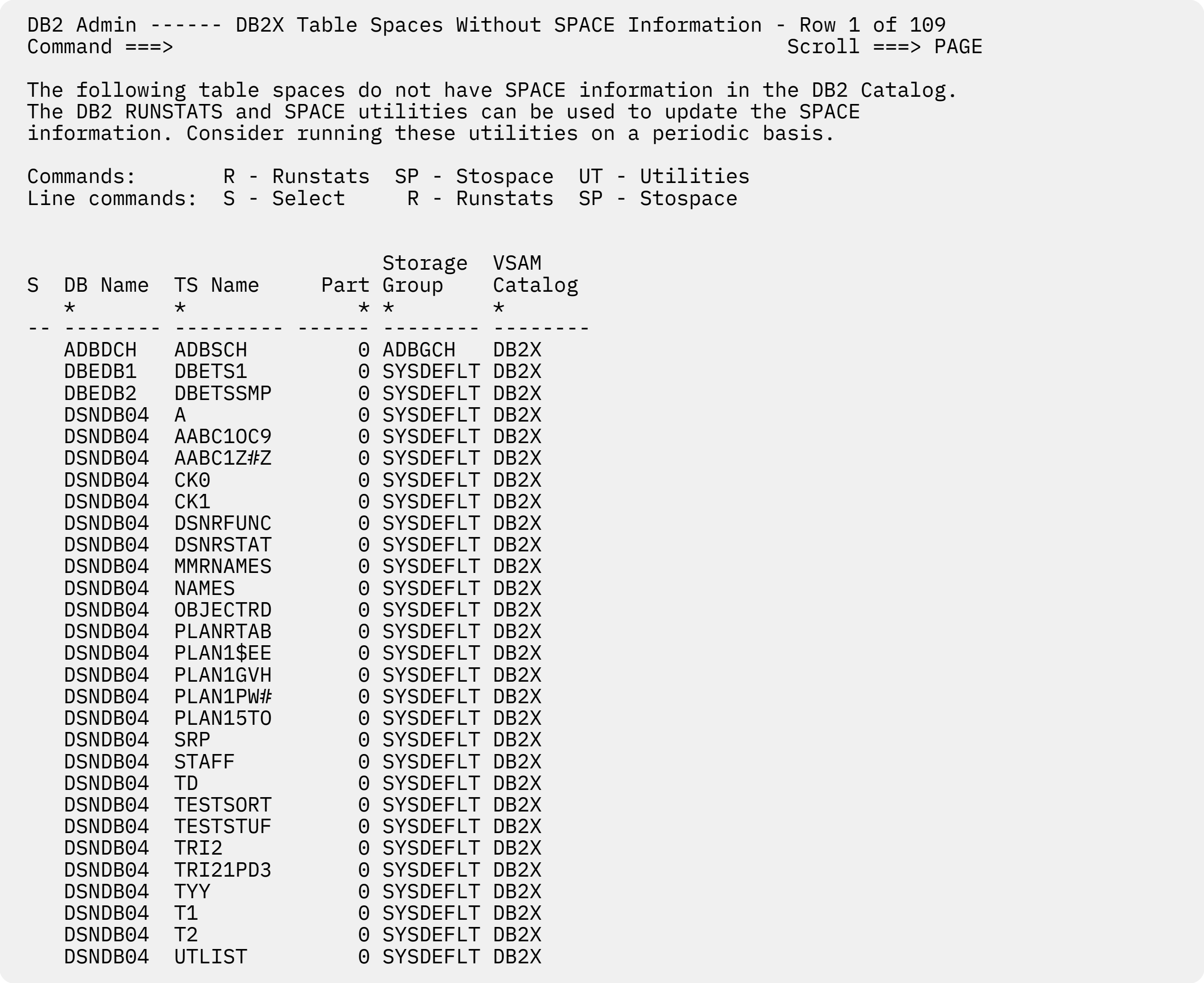
Use the performance queries function to view information about and work with table spaces that don't have SPACE information in the Db2 catalog.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Specify option 11, and press Enter.

   The **Table Spaces Without SPACE Information (ADB2311)** panel is displayed, as shown in the following figure:



```
DB2 Admin ------ DB2X Table Spaces Without SPACE Information - Row 1 of 109
Command ===>                                                  Scroll ===> PAGE

The following table spaces do not have SPACE information in the DB2 Catalog.
The DB2 RUNSTATS and SPACE utilities can be used to update the SPACE
information. Consider running these utilities on a periodic basis.

Commands:       R - Runstats  SP - Stospace  UT - Utilities
Line commands:  S - Select     R - Runstats  SP - Stospace


                             Storage  VSAM
S  DB Name  TS Name     Part Group    Catalog
   *        *              * *        *
-- -------- --------- ------ -------- --------
   ADBDCH   ADBSCH         0 ADBGCH   DB2X
   DBEDB1   DBETS1         0 SYSDEFLT DB2X
   DBEDB2   DBETSSMP       0 SYSDEFLT DB2X
   DSNDB04  A              0 SYSDEFLT DB2X
   DSNDB04  AABC1OC9       0 SYSDEFLT DB2X
   DSNDB04  AABC1Z#Z       0 SYSDEFLT DB2X
   DSNDB04  CK0            0 SYSDEFLT DB2X
   DSNDB04  CK1            0 SYSDEFLT DB2X
   DSNDB04  DSNRFUNC       0 SYSDEFLT DB2X
   DSNDB04  DSNRSTAT       0 SYSDEFLT DB2X
   DSNDB04  MMRNAMES       0 SYSDEFLT DB2X
   DSNDB04  NAMES          0 SYSDEFLT DB2X
   DSNDB04  OBJECTRD       0 SYSDEFLT DB2X
   DSNDB04  PLANRTAB       0 SYSDEFLT DB2X
   DSNDB04  PLAN1$EE       0 SYSDEFLT DB2X
   DSNDB04  PLAN1GVH       0 SYSDEFLT DB2X
   DSNDB04  PLAN1PW#       0 SYSDEFLT DB2X
   DSNDB04  PLAN15TO       0 SYSDEFLT DB2X
   DSNDB04  SRP            0 SYSDEFLT DB2X
   DSNDB04  STAFF          0 SYSDEFLT DB2X
   DSNDB04  TD             0 SYSDEFLT DB2X
   DSNDB04  TESTSORT       0 SYSDEFLT DB2X
   DSNDB04  TESTSTUF       0 SYSDEFLT DB2X
   DSNDB04  TRI2           0 SYSDEFLT DB2X
   DSNDB04  TRI21PD3       0 SYSDEFLT DB2X
   DSNDB04  TYY            0 SYSDEFLT DB2X
   DSNDB04  T1             0 SYSDEFLT DB2X
   DSNDB04  T2             0 SYSDEFLT DB2X
   DSNDB04  UTLIST         0 SYSDEFLT DB2X
```

*Figure 199. **Table Spaces Without SPACE Information (ADB2311)** panel*

The following columns are on this panel:

S
: Selects a table space when you use the S line command.

DB Name
: The name of the database on which the table resides.

TS Name
The name of the table on which the index is defined.

Part
The partition number. If it is not partitioned, the value is 0.

Storage Group
The name of the storage group for the table space.

VSAM Catalog
The name of the catalog that is used for space allocation.

3. Optional: Work with the table spaces by using the primary commands and line commands.

   For table spaces that do not have SPACE information in the Db2 catalog, use the Db2 RUNSTATS and STOSPACE utilities to update the SPACE information. Consider running these utilities on a periodic basis. You can run RUNSTATS with options that just update the SPACE fields in the catalog.

   The R line command enables you to quickly move to the **Batch Job Utility Parameters (ADB2UPA)** panel. Entering the R line command is equivalent to entering S, UT, and R commands in succession.

Running queries on indexes without SPACE information

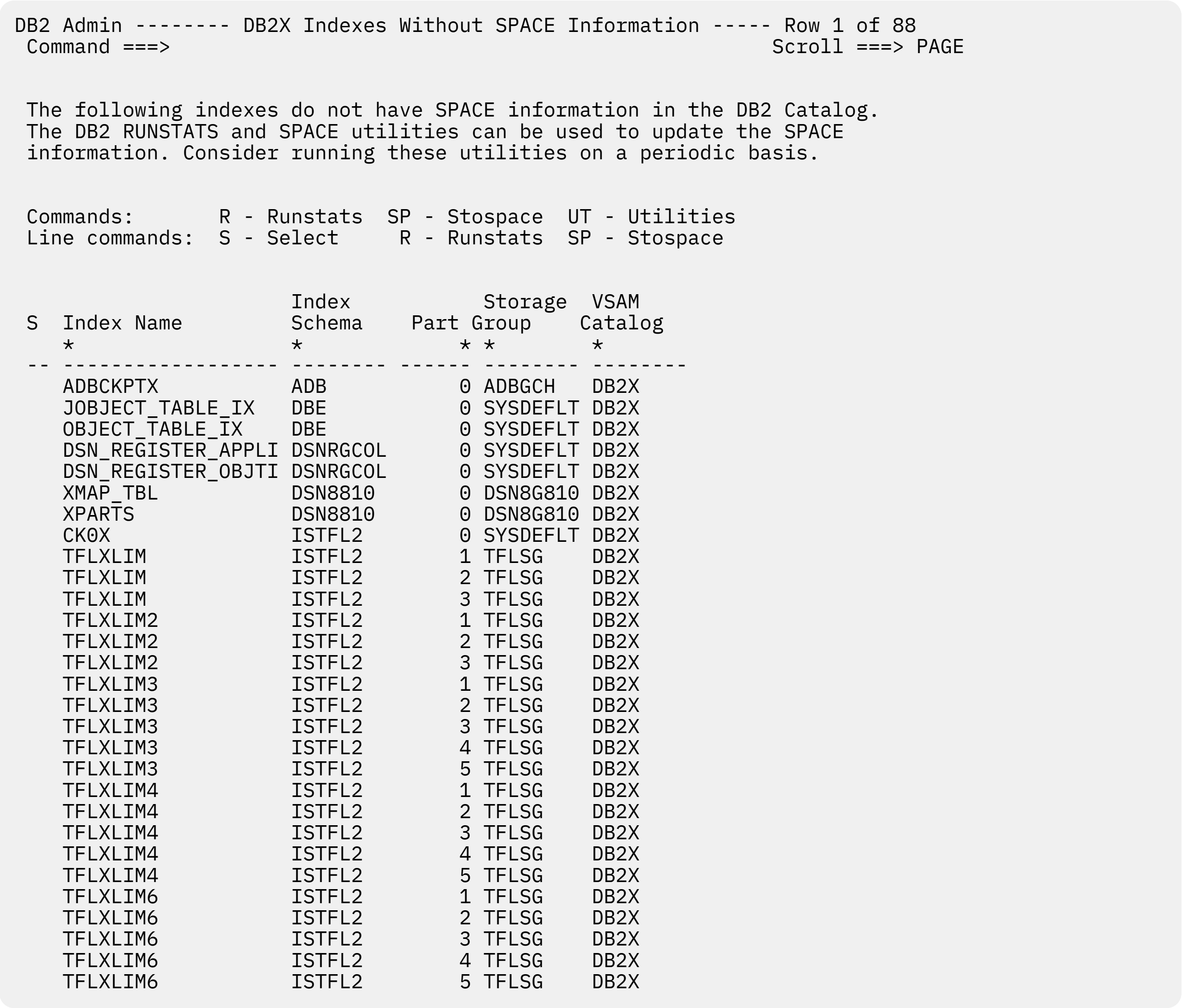
Use the performance queries function to view information about and work with indexes that don't have SPACE information in the Db2 catalog.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Specify option 11X, and press Enter.

   The **Indexes Without SPACE Information (ADB2311X)** panel is displayed, as shown in the following figure:



```
DB2 Admin -------- DB2X Indexes Without SPACE Information ----- Row 1 of 88
 Command ===>                                                  Scroll ===> PAGE


 The following indexes do not have SPACE information in the DB2 Catalog.
 The DB2 RUNSTATS and SPACE utilities can be used to update the SPACE
 information. Consider running these utilities on a periodic basis.


 Commands:       R - Runstats  SP - Stospace  UT - Utilities
 Line commands:  S - Select     R - Runstats  SP - Stospace


                       Index           Storage  VSAM
 S  Index Name         Schema    Part Group    Catalog
    *                  *             * *        *
 -- ------------------ -------- ------ -------- --------
    ADBCKPTX           ADB           0 ADBGCH   DB2X
    JOBJECT_TABLE_IX   DBE           0 SYSDEFLT DB2X
    OBJECT_TABLE_IX    DBE           0 SYSDEFLT DB2X
    DSN_REGISTER_APPLI DSNRGCOL      0 SYSDEFLT DB2X
    DSN_REGISTER_OBJTI DSNRGCOL      0 SYSDEFLT DB2X
    XMAP_TBL           DSN8810       0 DSN8G810 DB2X
    XPARTS             DSN8810       0 DSN8G810 DB2X
    CK0X               ISTFL2        0 SYSDEFLT DB2X
    TFLXLIM            ISTFL2        1 TFLSG    DB2X
    TFLXLIM            ISTFL2        2 TFLSG    DB2X
    TFLXLIM            ISTFL2        3 TFLSG    DB2X
    TFLXLIM2           ISTFL2        1 TFLSG    DB2X
    TFLXLIM2           ISTFL2        2 TFLSG    DB2X
    TFLXLIM2           ISTFL2        3 TFLSG    DB2X
    TFLXLIM3           ISTFL2        1 TFLSG    DB2X
    TFLXLIM3           ISTFL2        2 TFLSG    DB2X
    TFLXLIM3           ISTFL2        3 TFLSG    DB2X
    TFLXLIM3           ISTFL2        4 TFLSG    DB2X
    TFLXLIM3           ISTFL2        5 TFLSG    DB2X
    TFLXLIM4           ISTFL2        1 TFLSG    DB2X
    TFLXLIM4           ISTFL2        2 TFLSG    DB2X
    TFLXLIM4           ISTFL2        3 TFLSG    DB2X
    TFLXLIM4           ISTFL2        4 TFLSG    DB2X
    TFLXLIM4           ISTFL2        5 TFLSG    DB2X
    TFLXLIM6           ISTFL2        1 TFLSG    DB2X
    TFLXLIM6           ISTFL2        2 TFLSG    DB2X
    TFLXLIM6           ISTFL2        3 TFLSG    DB2X
    TFLXLIM6           ISTFL2        4 TFLSG    DB2X
    TFLXLIM6           ISTFL2        5 TFLSG    DB2X
```

Figure 200. ***Indexes Without SPACE Information (ADB2311X)*** *panel*

The following columns are on this panel:

S
Selects a table space when you use the S line command.

Index Name
The name of the index.

Index Schema
The authorization ID of the owner of the index.

Part
The partition number. If it is not partitioned, the value is 0.

Storage Group
The name of the storage group for the index.

VSAM Catalog
The name of the catalog that is used for space allocation.

3. Optional: Work with the indexes by using the primary commands and line commands.

For indexes that do not have SPACE information in the Db2 catalog, use the Db2 RUNSTATS and SPACE utilities to update the SPACE information. Consider running these utilities on a periodic basis.

The R line command enables you to quickly move to the **Batch Job Utility Parameters (ADB2UPA)** panel. Entering the R line command is equivalent to entering S, UT, and R commands in succession.

Running queries on table spaces exceeding allocated primary quantity

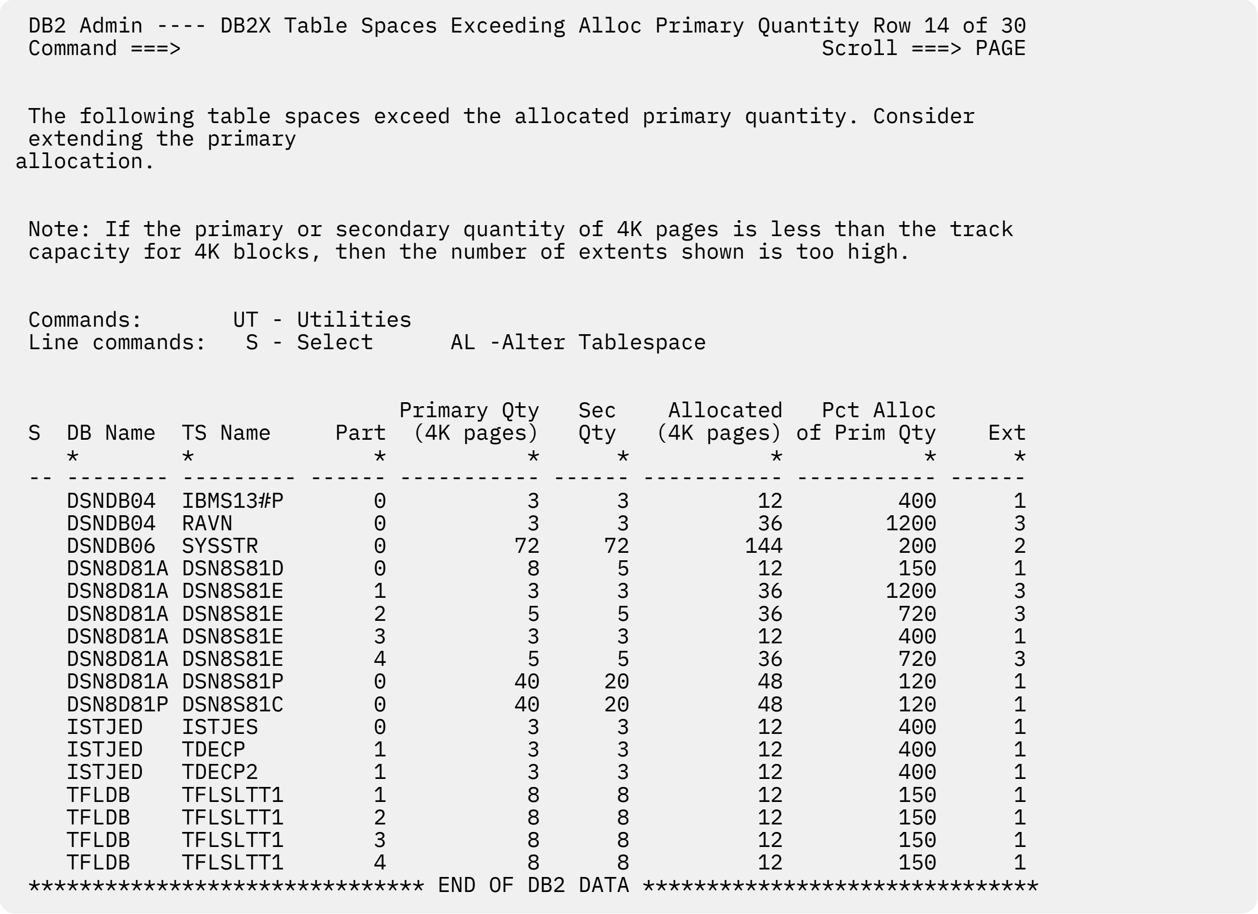
Use the performance queries function to view information about and work with tables spaces that exceed the allocated primary quantity.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Specify option 12, and press Enter.

   The **Table Spaces Exceeding Allocated Primary Quantity (ADB2312)** panel is displayed, as shown in the following figure:



```
 DB2 Admin ---- DB2X Table Spaces Exceeding Alloc Primary Quantity Row 14 of 30
 Command ===>                                                  Scroll ===> PAGE


 The following table spaces exceed the allocated primary quantity. Consider
 extending the primary
allocation.


 Note: If the primary or secondary quantity of 4K pages is less than the track
 capacity for 4K blocks, then the number of extents shown is too high.


 Commands:       UT - Utilities
 Line commands:   S - Select      AL -Alter Tablespace


                             Primary Qty   Sec    Allocated   Pct Alloc
 S  DB Name  TS Name     Part  (4K pages)   Qty   (4K pages) of Prim Qty    Ext
    *        *              *           *      *           *           *      *
 -- -------- --------- ------ ----------- ------ ----------- ----------- ------
    DSNDB04  IBMS13#P       0           3      3          12         400      1
    DSNDB04  RAVN           0           3      3          36        1200      3
    DSNDB06  SYSSTR         0          72     72         144         200      2
    DSN8D81A DSN8S81D       0           8      5          12         150      1
    DSN8D81A DSN8S81E       1           3      3          36        1200      3
    DSN8D81A DSN8S81E       2           5      5          36         720      3
    DSN8D81A DSN8S81E       3           3      3          12         400      1
    DSN8D81A DSN8S81E       4           5      5          36         720      3
    DSN8D81A DSN8S81P       0          40     20          48         120      1
    DSN8D81P DSN8S81C       0          40     20          48         120      1
    ISTJED   ISTJES         0           3      3          12         400      1
    ISTJED   TDECP          1           3      3          12         400      1
    ISTJED   TDECP2         1           3      3          12         400      1
    TFLDB    TFLSLTT1       1           8      8          12         150      1
    TFLDB    TFLSLTT1       2           8      8          12         150      1
    TFLDB    TFLSLTT1       3           8      8          12         150      1
    TFLDB    TFLSLTT1       4           8      8          12         150      1
 ******************************* END OF DB2 DATA *******************************
```

*Figure 201. **Table Spaces Exceeding Allocated Primary Quantity (ADB2312)** panel*

The following columns are on this panel:

S
Selects a table space when you use the S line command.

DB Name
The name of the database.

TS Name
The name of the table space.

Part
The partition number. If it is not partitioned, the value is 0.

Primary Qty (4K pages)
Primary quantity. This column shows the primary space allocation in 4K blocks of storage.

Sec Qty
: Secondary quantity. This column shows the secondary space allocation in 4K blocks of storage.

Allocated (4K pages)
: The allocated space in 4K blocks of storage.

Pct Alloc of Prim Qty
: The percentage of the primary quantity of space that is allocated.

Ext
: The estimated number of extents for the table space.

3. Optional: Work with the table spaces by using the primary commands and line commands.

   For table spaces that exceed the allocated primary quantity, consider extending the primary allocation.

   The AL line command enables you to quickly move to the **Alter Table Space (ADB21SA)** panel. Entering the AL line command is equivalent to entering an S line command and then entering an AL line command.

Running queries on indexes exceeding allocated primary quantity

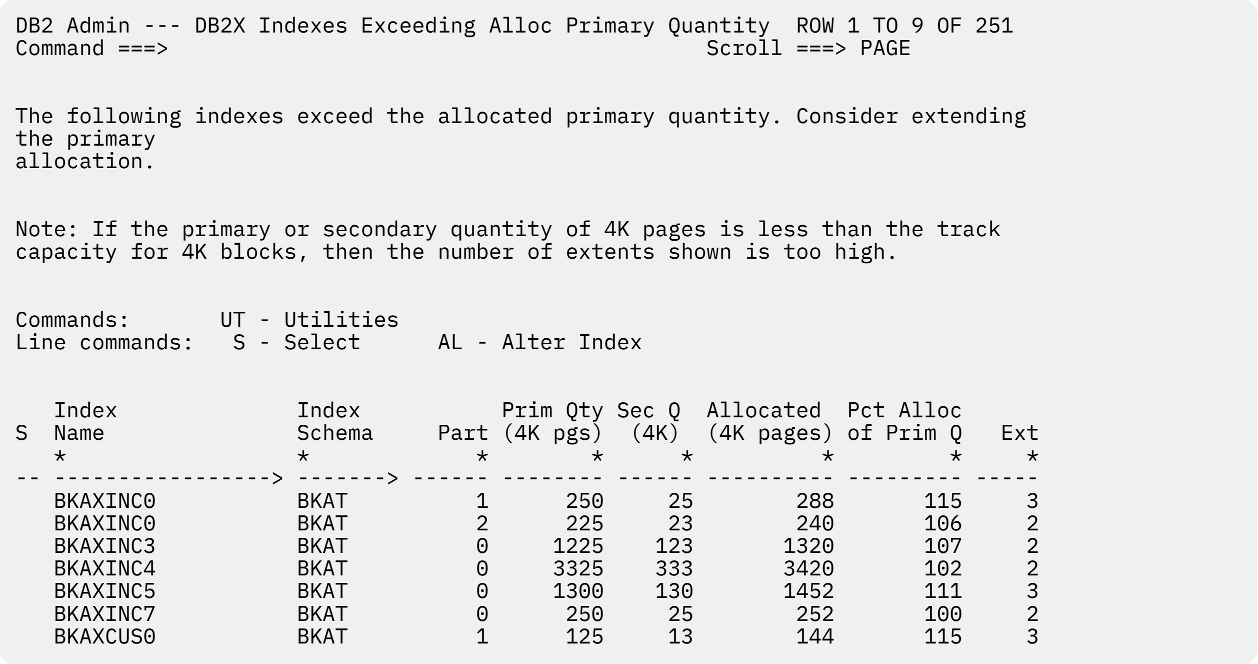
Use the performance queries function to view information about and work with indexes that exceed the allocated primary quantity.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Specify option 12X, and press Enter.

   The **Indexes Exceeding Alloc Primary Quantity (ADB2312X)** panel is displayed, as shown in the following figure:



```
DB2 Admin --- DB2X Indexes Exceeding Alloc Primary Quantity  ROW 1 TO 9 OF 251
Command ===>                                          Scroll ===> PAGE


The following indexes exceed the allocated primary quantity. Consider extending
the primary
allocation.


Note: If the primary or secondary quantity of 4K pages is less than the track
capacity for 4K blocks, then the number of extents shown is too high.


Commands:       UT - Utilities
Line commands:   S - Select      AL - Alter Index


   Index              Index           Prim Qty Sec Q  Allocated  Pct Alloc
S  Name               Schema     Part (4K pgs)  (4K)  (4K pages) of Prim Q   Ext
   *                  *             *        *      *          *         *     *
-- -----------------> -------> ------ -------- ------ ---------- --------- -----
   BKAXINC0           BKAT          1      250     25        288       115     3
   BKAXINC0           BKAT          2      225     23        240       106     2
   BKAXINC3           BKAT          0     1225    123       1320       107     2
   BKAXINC4           BKAT          0     3325    333       3420       102     2
   BKAXINC5           BKAT          0     1300    130       1452       111     3
   BKAXINC7           BKAT          0      250     25        252       100     2
   BKAXCUS0           BKAT          1      125     13        144       115     3
```

*Figure 202. **Indexes Exceeding Alloc Primary Quantity (ADB2312X)** panel*

The following columns are on this panel:

S
: Selects a table space when you use the S line command.

Index Name
The name of the index.

Index Schema
The authorization ID of the owner of the index.

Part
The partition number. If it is not partitioned, the value is 0.

Primary Qty (4K pages)
Primary quantity. This column shows the primary space allocation in 4K blocks of storage.

Sec Qty
Secondary quantity. This column shows the secondary space allocation in 4K blocks of storage.

Allocated (4K pages)
The allocated space in 4K blocks of storage.

Pct Alloc of Prim Qty
The percentage of the primary quantity of space that is allocated.

Ext
The estimated number of extents for the index.

3. Optional: Work with the indexes by using the primary commands and line commands.

   For indexes that exceed the allocated primary quantity, consider extending the primary allocation.

   The AL line command enables you to quickly move to the **Alter Index (ADB21XA)** panel. Entering the AL line command is equivalent to entering an S line command and then entering an AL line command.

Running queries on allocated and used space for table spaces

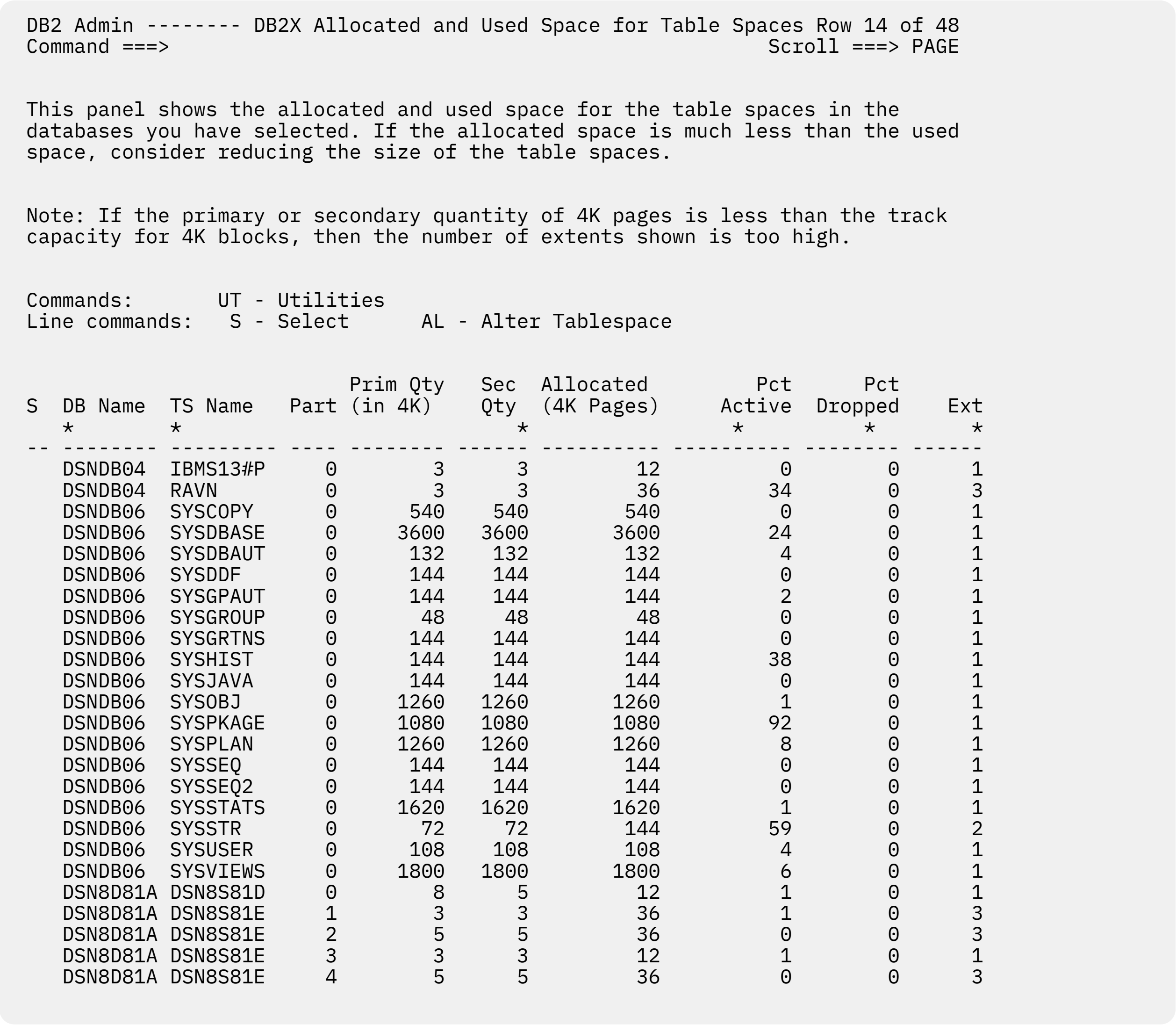
Use the performance queries function to view information about and work with indexes that exceed the allocated primary quantity.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Specify option 13, and press Enter.

   The **Allocated and Used Space for Table Spaces (ADB2313)** panel is displayed, as shown in the following figure:



```
DB2 Admin -------- DB2X Allocated and Used Space for Table Spaces Row 14 of 48
Command ===>                                                  Scroll ===> PAGE


This panel shows the allocated and used space for the table spaces in the
databases you have selected. If the allocated space is much less than the used
space, consider reducing the size of the table spaces.


Note: If the primary or secondary quantity of 4K pages is less than the track
capacity for 4K blocks, then the number of extents shown is too high.


Commands:       UT - Utilities
Line commands:   S - Select      AL - Alter Tablespace


                           Prim Qty   Sec  Allocated         Pct       Pct
S  DB Name  TS Name   Part (in 4K)    Qty  (4K Pages)     Active  Dropped    Ext
   *        *                            *                 *          *        *
-- -------- --------- ---- -------- ------ ---------- ---------- -------- ------
   DSNDB04  IBMS13#P     0        3      3          12          0        0      1
   DSNDB04  RAVN         0        3      3          36         34        0      3
   DSNDB06  SYSCOPY      0      540    540         540          0        0      1
   DSNDB06  SYSDBASE     0     3600   3600        3600         24        0      1
   DSNDB06  SYSDBAUT     0      132    132         132          4        0      1
   DSNDB06  SYSDDF       0      144    144         144          0        0      1
   DSNDB06  SYSGPAUT     0      144    144         144          2        0      1
   DSNDB06  SYSGROUP     0       48     48          48          0        0      1
   DSNDB06  SYSGRTNS     0      144    144         144          0        0      1
   DSNDB06  SYSHIST      0      144    144         144         38        0      1
   DSNDB06  SYSJAVA      0      144    144         144          0        0      1
   DSNDB06  SYSOBJ       0     1260   1260        1260          1        0      1
   DSNDB06  SYSPKAGE     0     1080   1080        1080         92        0      1
   DSNDB06  SYSPLAN      0     1260   1260        1260          8        0      1
   DSNDB06  SYSSEQ       0      144    144         144          0        0      1
   DSNDB06  SYSSEQ2      0      144    144         144          0        0      1
   DSNDB06  SYSSTATS     0     1620   1620        1620          1        0      1
   DSNDB06  SYSSTR       0       72     72         144         59        0      2
   DSNDB06  SYSUSER      0      108    108         108          4        0      1
   DSNDB06  SYSVIEWS     0     1800   1800        1800          6        0      1
   DSN8D81A DSN8S81D     0        8      5          12          1        0      1
   DSN8D81A DSN8S81E     1        3      3          36          1        0      3
   DSN8D81A DSN8S81E     2        5      5          36          0        0      3
   DSN8D81A DSN8S81E     3        3      3          12          1        0      1
   DSN8D81A DSN8S81E     4        5      5          36          0        0      3
```

Figure 203. ***Allocated and Used Space for Table Spaces (ADB2313)*** *panel*

The following columns are on this panel:

S
Selects a table space when you use the S line command.

DB Name
The name of database.

TS Name
The name of the table space.

Part
The partition number. If it is not partitioned, the value is 0.

Prim Qty (in 4K)
Primary quantity. This column shows the primary space allocation in 4K blocks of storage.

Sec Qty
Secondary quantity. This column shows the secondary space allocation in 4K blocks of storage.

Allocated (4K Pages)
The allocated space in 4K blocks of storage.

Pct Active
The percentage of the space that is occupied by rows of data from active tables.

Pct Dropped
The percentage of the space this is occupied by rows of data from dropped tables.

Ext
The estimated number of extents for the table space.

3. Optional: Work with the indexes by using the primary commands and line commands.

   The **DB2 Performance Queries** panel shows the allocated and used space for the table spaces in the databases you have selected. If the allocated space is much less than the used space, consider reducing the size of the table spaces.

   The AL line command enables you to quickly move to the **Alter Table Space (ADB21SA)** panel. Entering the AL line command is equivalent to entering an S line command and then entering an AL line command.

Requesting table space maintenance recommendations

Db2 Admin Tool can use data from the real-time statistics (RTS) tables to provide recommendations on when to run certain maintenance functions, such as COPY, REORG, or RUNSTATS, on your table spaces.

Before you begin

To get these table space recommendations, real-time statistics tables must exist.

Procedure

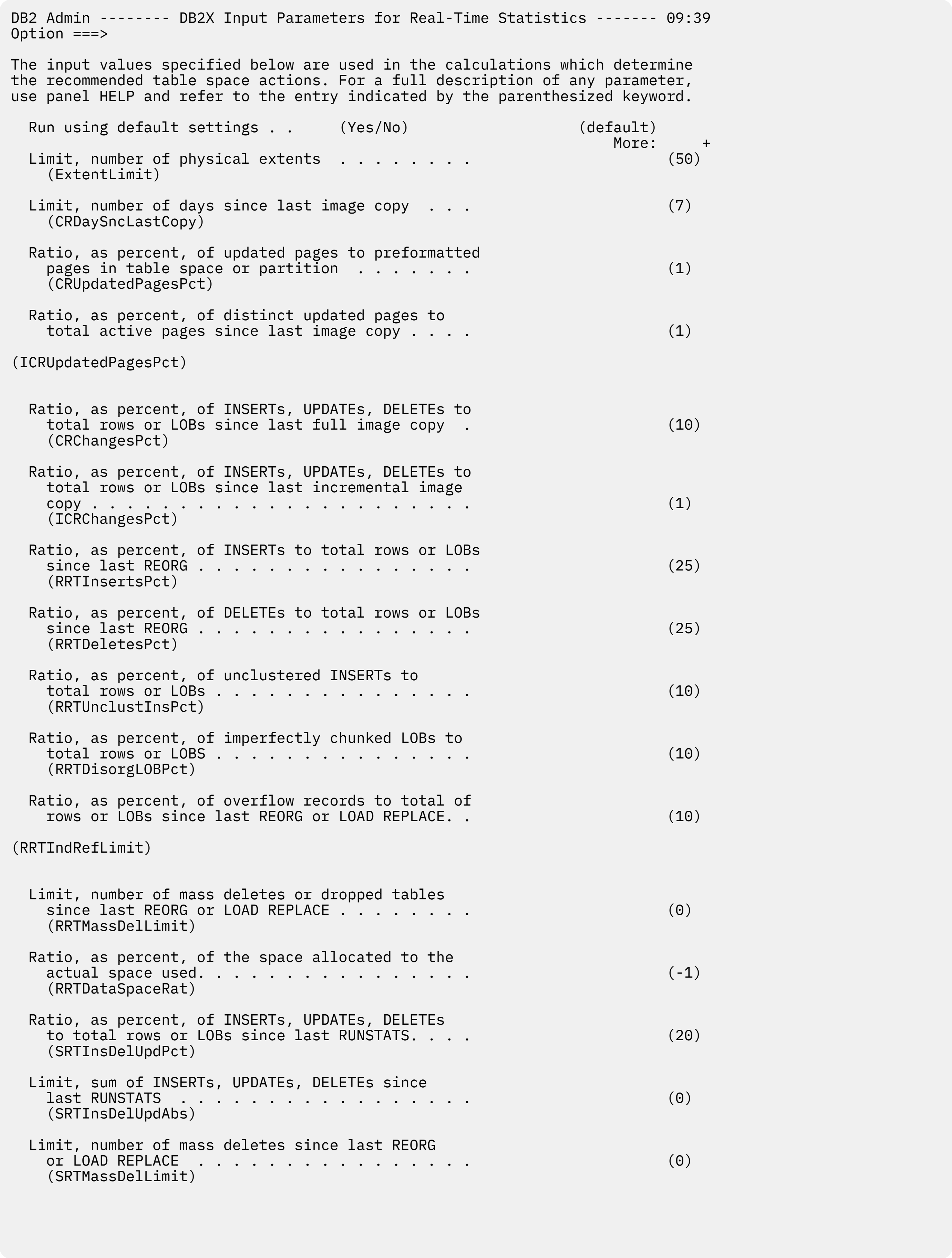
To request table space maintenance recommendations:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.
2. On the **DB2 Performance Queries (ADB23)** panel, specify option 14, and press Enter.
3. On the **Input Parameters for Real-Time Statistics (ADB2314T)** panel, specify your own values for the fields or use the system default values, and press Enter.

   These values are used to calculate recommendations that can help you to determine when to run certain maintenance functions or when to enlarge your Db2 data sets.

   Important: The recommendations that Db2 Admin Tool provides are based on general formulas and might not apply or be accurate for every installation. Additionally, if the real-time statistics tables contain only a small portion of information about your Db2 subsystem, the recommendations might not apply to the entire subsystem.

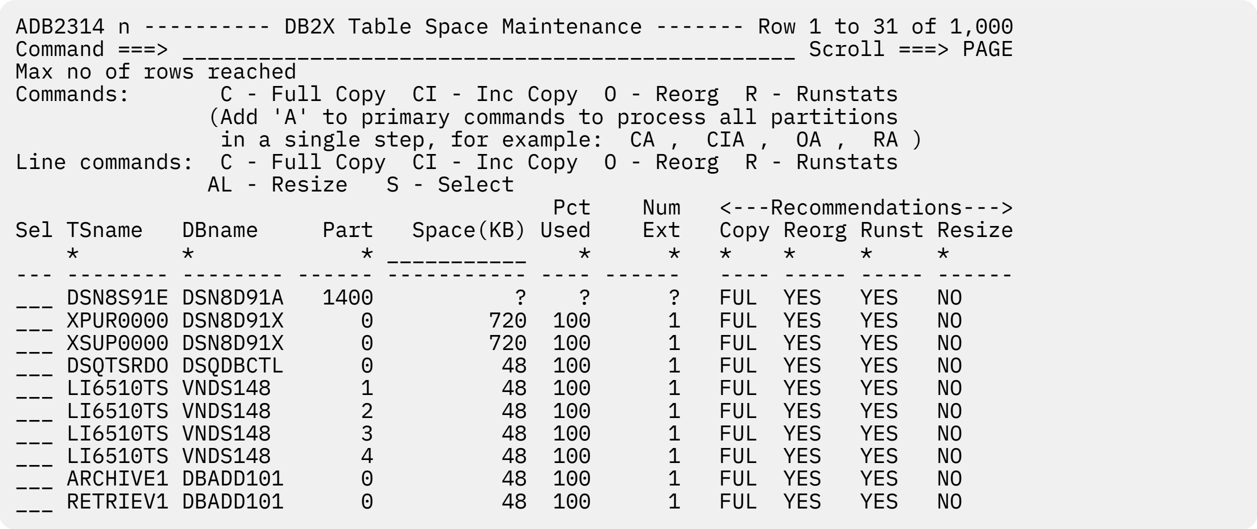
   To reset all user values to the system default values, issue the RESET primary command, and press Enter.



```
DB2 Admin -------- DB2X Input Parameters for Real-Time Statistics ------- 09:39
Option ===>

The input values specified below are used in the calculations which determine
the recommended table space actions. For a full description of any parameter,
use panel HELP and refer to the entry indicated by the parenthesized keyword.

  Run using default settings . .     (Yes/No)                   (default)
                                                                    More:     +
  Limit, number of physical extents  . . . . . . . .                      (50)
    (ExtentLimit)

  Limit, number of days since last image copy  . . .                      (7)
    (CRDaySncLastCopy)

  Ratio, as percent, of updated pages to preformatted
    pages in table space or partition  . . . . . . .                      (1)
    (CRUpdatedPagesPct)

  Ratio, as percent, of distinct updated pages to
    total active pages since last image copy . . . .                      (1)

(ICRUpdatedPagesPct)


  Ratio, as percent, of INSERTs, UPDATEs, DELETEs to
    total rows or LOBs since last full image copy  .                      (10)
    (CRChangesPct)

  Ratio, as percent, of INSERTs, UPDATEs, DELETEs to
    total rows or LOBs since last incremental image
    copy . . . . . . . . . . . . . . . . . . . . . .                      (1)
    (ICRChangesPct)

  Ratio, as percent, of INSERTs to total rows or LOBs
    since last REORG . . . . . . . . . . . . . . . .                      (25)
    (RRTInsertsPct)

  Ratio, as percent, of DELETEs to total rows or LOBs
    since last REORG . . . . . . . . . . . . . . . .                      (25)
    (RRTDeletesPct)

  Ratio, as percent, of unclustered INSERTs to
    total rows or LOBs . . . . . . . . . . . . . . .                      (10)
    (RRTUnclustInsPct)

  Ratio, as percent, of imperfectly chunked LOBs to
    total rows or LOBS . . . . . . . . . . . . . . .                      (10)
    (RRTDisorgLOBPct)

  Ratio, as percent, of overflow records to total of
    rows or LOBs since last REORG or LOAD REPLACE. .                      (10)

(RRTIndRefLimit)


  Limit, number of mass deletes or dropped tables
    since last REORG or LOAD REPLACE . . . . . . . .                      (0)
    (RRTMassDelLimit)

  Ratio, as percent, of the space allocated to the
    actual space used. . . . . . . . . . . . . . . .                      (-1)
    (RRTDataSpaceRat)

  Ratio, as percent, of INSERTs, UPDATEs, DELETEs
    to total rows or LOBs since last RUNSTATS. . . .                      (20)
    (SRTInsDelUpdPct)

  Limit, sum of INSERTs, UPDATEs, DELETEs since
    last RUNSTATS  . . . . . . . . . . . . . . . . .                      (0)
    (SRTInsDelUpdAbs)

  Limit, number of mass deletes since last REORG
    or LOAD REPLACE  . . . . . . . . . . . . . . . .                      (0)
    (SRTMassDelLimit)
```

Figure 204. ***Input Parameters for Real-Time Statistics (ADB2314T)*** *panel*

When you press Enter, recommendations are displayed, as shown in the following example **Table Space Maintenance (ADB2314)** panel:



```
ADB2314 n ---------- DB2X Table Space Maintenance ------- Row 1 to 31 of 1,000
Command ===> ________________________________________________ Scroll ===> PAGE
Max no of rows reached
Commands:       C - Full Copy  CI - Inc Copy  O - Reorg  R - Runstats
               (Add 'A' to primary commands to process all partitions
                in a single step, for example:  CA ,  CIA ,  OA ,  RA )
Line commands:  C - Full Copy  CI - Inc Copy  O - Reorg  R - Runstats
               AL - Resize   S - Select
                                          Pct    Num   <---Recommendations--->
Sel TSname   DBname     Part   Space(KB) Used    Ext   Copy Reorg Runst Resize
    *        *             * ___________    *      *   *    *     *     *
--- -------- -------- ------ ----------- ---- ------   ---- ----- ----- ------
___ DSN8S91E DSN8D91A   1400           ?    ?      ?   FUL  YES   YES   NO
___ XPUR0000 DSN8D91X      0         720  100      1   FUL  YES   YES   NO
___ XSUP0000 DSN8D91X      0         720  100      1   FUL  YES   YES   NO
___ DSQTSRDO DSQDBCTL      0          48  100      1   FUL  YES   YES   NO
___ LI6510TS VNDS148       1          48  100      1   FUL  YES   YES   NO
___ LI6510TS VNDS148       2          48  100      1   FUL  YES   YES   NO
___ LI6510TS VNDS148       3          48  100      1   FUL  YES   YES   NO
___ LI6510TS VNDS148       4          48  100      1   FUL  YES   YES   NO
___ ARCHIVE1 DBADD101      0          48  100      1   FUL  YES   YES   NO
___ RETRIEV1 DBADD101      0          48  100      1   FUL  YES   YES   NO
```

*Figure 205. **Table Space Maintenance (ADB2314)** panel, which is the result of panel ADB2314T*

Requesting index space maintenance recommendations

Db2 Admin Tool can use data from the real-time statistics (RTS) tables to provide recommendations on when to run certain maintenance functions, such as COPY, REORG, or RUNSTATS, on your index spaces.

Before you begin

To get these index space recommendations, real-time statistics tables must exist.

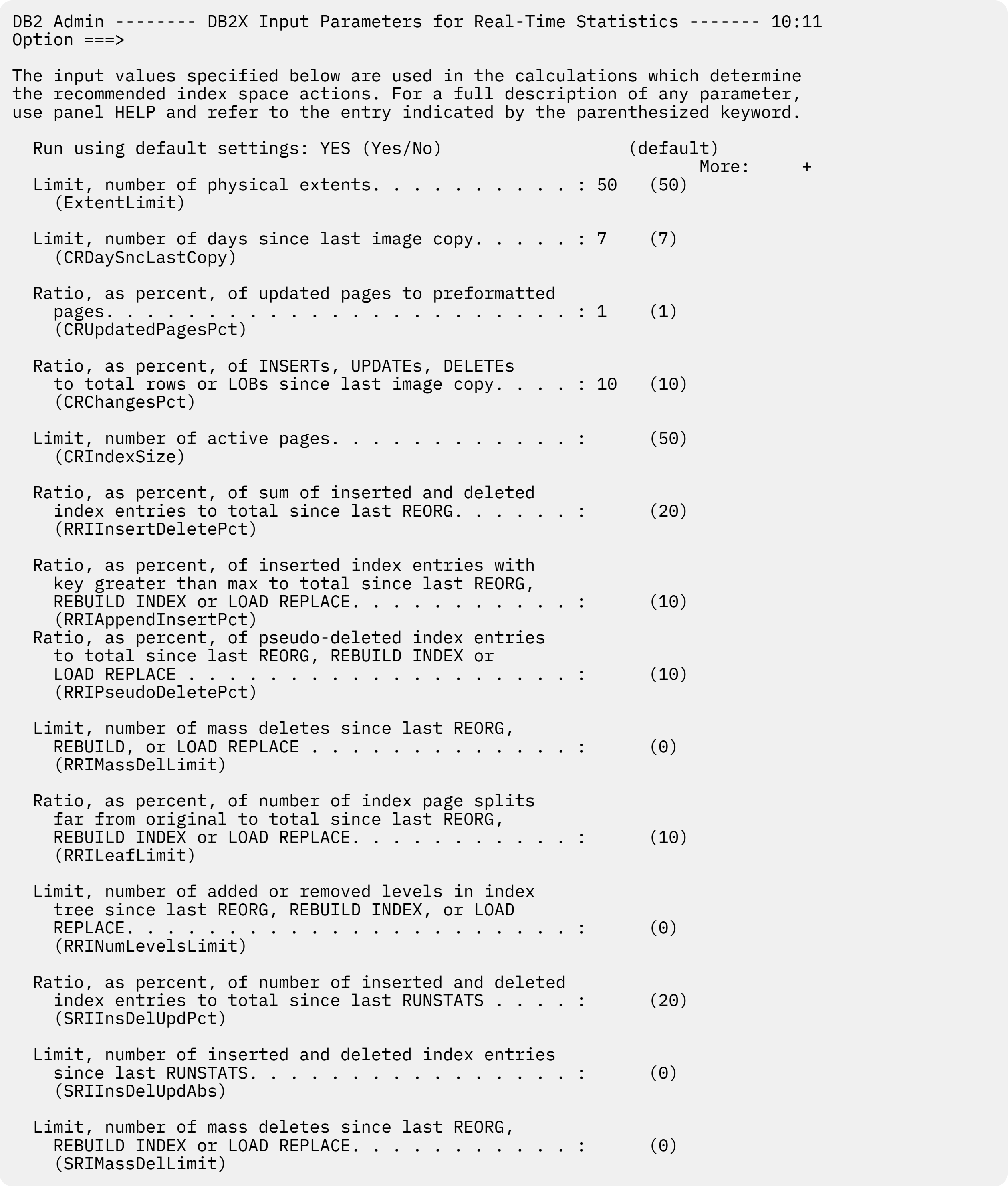
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.
2. On the **DB2 Performance Queries (ADB23)** panel, specify option 14X, and press Enter.
3. On the **Input Parameters for Real-Time Statistics (ADB2314I)** panel, specify your own values for the fields or use the system default values, and press Enter.

   These values are used to calculate recommendations that can help you to determine when to run certain maintenance functions or when to enlarge your Db2 data sets.

   Important: The recommendations that Db2 Admin Tool provides are based on general formulas and might not apply or be accurate for every installation. Additionally, if the real-time statistics tables contain only a small portion of information about your Db2 subsystem, the recommendations might not apply to the entire subsystem.

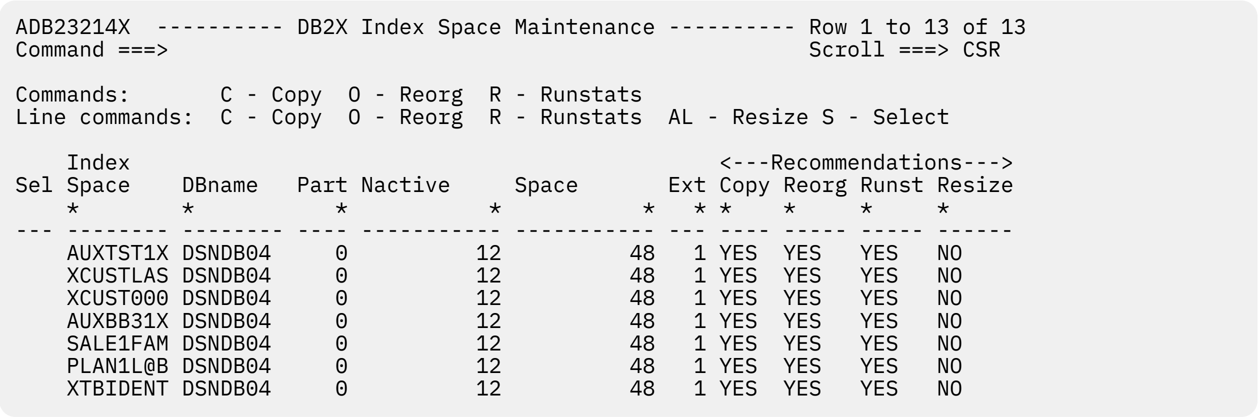
   To reset all user values to the system default values, issue the RESET primary command, and press Enter.



```
DB2 Admin -------- DB2X Input Parameters for Real-Time Statistics ------- 10:11
Option ===>

The input values specified below are used in the calculations which determine
the recommended index space actions. For a full description of any parameter,
use panel HELP and refer to the entry indicated by the parenthesized keyword.

  Run using default settings: YES (Yes/No)                  (default)
                                                                   More:     +
  Limit, number of physical extents. . . . . . . . . . : 50   (50)
    (ExtentLimit)

  Limit, number of days since last image copy. . . . . : 7    (7)
    (CRDaySncLastCopy)

  Ratio, as percent, of updated pages to preformatted
    pages. . . . . . . . . . . . . . . . . . . . . . . . : 1    (1)
    (CRUpdatedPagesPct)

  Ratio, as percent, of INSERTs, UPDATEs, DELETEs
    to total rows or LOBs since last image copy. . . . : 10   (10)
    (CRChangesPct)

  Limit, number of active pages. . . . . . . . . . . . :      (50)
    (CRIndexSize)

  Ratio, as percent, of sum of inserted and deleted
    index entries to total since last REORG. . . . . . :      (20)
    (RRIInsertDeletePct)

  Ratio, as percent, of inserted index entries with
    key greater than max to total since last REORG,
    REBUILD INDEX or LOAD REPLACE. . . . . . . . . . . . :      (10)
    (RRIAppendInsertPct)
  Ratio, as percent, of pseudo-deleted index entries
    to total since last REORG, REBUILD INDEX or
    LOAD REPLACE . . . . . . . . . . . . . . . . . . . . :      (10)
    (RRIPseudoDeletePct)

  Limit, number of mass deletes since last REORG,
    REBUILD, or LOAD REPLACE . . . . . . . . . . . . . . :      (0)
    (RRIMassDelLimit)

  Ratio, as percent, of number of index page splits
    far from original to total since last REORG,
    REBUILD INDEX or LOAD REPLACE. . . . . . . . . . . . :      (10)
    (RRILeafLimit)

  Limit, number of added or removed levels in index
    tree since last REORG, REBUILD INDEX, or LOAD
    REPLACE. . . . . . . . . . . . . . . . . . . . . . . :      (0)
    (RRINumLevelsLimit)

  Ratio, as percent, of number of inserted and deleted
    index entries to total since last RUNSTATS . . . . . :      (20)
    (SRIInsDelUpdPct)

  Limit, number of inserted and deleted index entries
    since last RUNSTATS. . . . . . . . . . . . . . . . . :      (0)
    (SRIInsDelUpdAbs)

  Limit, number of mass deletes since last REORG,
    REBUILD INDEX or LOAD REPLACE. . . . . . . . . . . . :      (0)
    (SRIMassDelLimit)
```

*Figure 206. **Input Parameters for Real-Time Statistics (ADB2314I)** panel*

When you press Enter, recommendations are displayed, as shown in the following example **Index Space Maintenance (ADB2314X)** panel:



```
ADB23214X  ---------- DB2X Index Space Maintenance ---------- Row 1 to 13 of 13
Command ===>                                                  Scroll ===> CSR

Commands:       C - Copy  O - Reorg  R - Runstats
Line commands:  C - Copy  O - Reorg  R - Runstats  AL - Resize S - Select

    Index                                              <---Recommendations--->
Sel Space    DBname   Part Nactive     Space       Ext Copy Reorg Runst Resize
    *        *           *           *           *   * *    *     *     *
--- -------- -------- ---- ----------- ----------- --- ---- ----- ----- ------
    AUXTST1X DSNDB04     0          12          48   1 YES  YES   YES   NO
    XCUSTLAS DSNDB04     0          12          48   1 YES  YES   YES   NO
    XCUST000 DSNDB04     0          12          48   1 YES  YES   YES   NO
    AUXBB31X DSNDB04     0          12          48   1 YES  YES   YES   NO
    SALE1FAM DSNDB04     0          12          48   1 YES  YES   YES   NO
    PLAN1L@B DSNDB04     0          12          48   1 YES  YES   YES   NO
    XTBIDENT DSNDB04     0          12          48   1 YES  YES   YES   NO
```

Figure 207. ***Index Space Maintenance (ADB2314X)*** *panel, which is the result of panel ADB2314I*

Running queries on indexes not used within *n* number of days

Use the performance queries function to view information about and work with indexes that have not been used within a specified number of days.

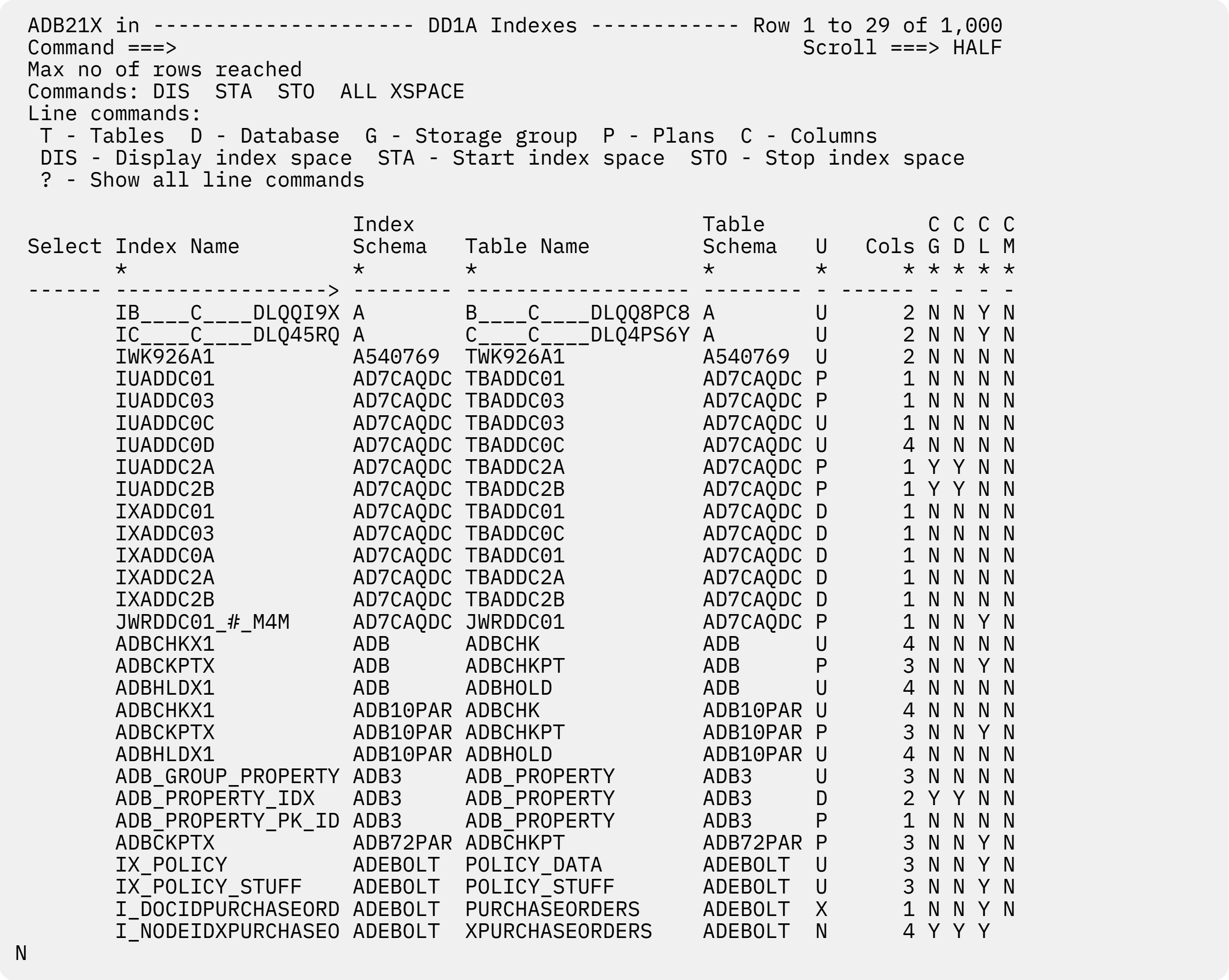
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 3, and press Enter.

   The **DB2 Performance Queries** panel is displayed.
2. Locate option **15 - Indexes not used within *n* days**.
3. If necessary, change the value of *n*.

   Valid values are 1 - 99999.
4. Specify option 15, and press Enter.

   The **Indexes (ADB21X)** panel is displayed, as shown in the following figure:



```
 ADB21X in -------------------- DD1A Indexes ------------ Row 1 to 29 of 1,000
 Command ===>                                                 Scroll ===> HALF
 Max no of rows reached
 Commands: DIS  STA  STO  ALL XSPACE
 Line commands:
  T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
  DIS - Display index space  STA - Start index space  STO - Stop index space
  ? - Show all line commands

                           Index                       Table                C C C C
 Select Index Name         Schema   Table Name         Schema   U   Cols G D L M
        *                  *        *                  *        *      * * * * *
 ------ -----------------> -------- ------------------ -------- - ------ - - - -
        IB____C____DLQQI9X A        B____C____DLQQ8PC8 A        U      2 N N Y N
        IC____C____DLQ45RQ A        C____C____DLQ4PS6Y A        U      2 N N Y N
        IWK926A1           A540769  TWK926A1           A540769  U      2 N N N N
        IUADDC01           AD7CAQDC TBADDC01           AD7CAQDC P      1 N N N N
        IUADDC03           AD7CAQDC TBADDC03           AD7CAQDC P      1 N N N N
        IUADDC0C           AD7CAQDC TBADDC03           AD7CAQDC U      1 N N N N
        IUADDC0D           AD7CAQDC TBADDC0C           AD7CAQDC U      4 N N N N
        IUADDC2A           AD7CAQDC TBADDC2A           AD7CAQDC P      1 Y Y N N
        IUADDC2B           AD7CAQDC TBADDC2B           AD7CAQDC P      1 Y Y N N
        IXADDC01           AD7CAQDC TBADDC01           AD7CAQDC D      1 N N N N
        IXADDC03           AD7CAQDC TBADDC0C           AD7CAQDC D      1 N N N N
        IXADDC0A           AD7CAQDC TBADDC01           AD7CAQDC D      1 N N N N
        IXADDC2A           AD7CAQDC TBADDC2A           AD7CAQDC D      1 N N N N
        IXADDC2B           AD7CAQDC TBADDC2B           AD7CAQDC D      1 N N N N
        JWRDDC01_#_M4M     AD7CAQDC JWRDDC01           AD7CAQDC P      1 N N Y N
        ADBCHKX1           ADB      ADBCHK             ADB      U      4 N N N N
        ADBCKPTX           ADB      ADBCHKPT           ADB      P      3 N N Y N
        ADBHLDX1           ADB      ADBHOLD            ADB      U      4 N N N N
        ADBCHKX1           ADB10PAR ADBCHK             ADB10PAR U      4 N N N N
        ADBCKPTX           ADB10PAR ADBCHKPT           ADB10PAR P      3 N N Y N
        ADBHLDX1           ADB10PAR ADBHOLD            ADB10PAR U      4 N N N N
        ADB_GROUP_PROPERTY ADB3     ADB_PROPERTY       ADB3     U      3 N N N N
        ADB_PROPERTY_IDX   ADB3     ADB_PROPERTY       ADB3     D      2 Y Y N N
        ADB_PROPERTY_PK_ID ADB3     ADB_PROPERTY       ADB3     P      1 N N N N
        ADBCKPTX           ADB72PAR ADBCHKPT           ADB72PAR P      3 N N Y N
        IX_POLICY          ADEBOLT  POLICY_DATA        ADEBOLT  U      3 N N Y N
        IX_POLICY_STUFF    ADEBOLT  POLICY_STUFF       ADEBOLT  U      3 N N Y N
        I_DOCIDPURCHASEORD ADEBOLT  PURCHASEORDERS     ADEBOLT  X      1 N N Y N
        I_NODEIDXPURCHASEO ADEBOLT  XPURCHASEORDERS    ADEBOLT  N      4 Y Y Y
N
```

Figure 208. ***Indexes (ADB21X)*** *panel*

The following columns are on this panel:

Select
: Selects a table space when you use the S line command.

Index Name
: The name of the index.

Index Schema
: The authorization ID of the schema of the index.

Table Name
: The name of the table on which the index is defined.

Table Schema
: The authorization ID of the schema of the table.

5. Optional: Work with the indexes by using the primary commands and line commands.

LISTDEF and TEMPLATE

LISTDEF and TEMPLATE are Db2 utilities that provide facilities for other utilities. You can use LISTDEF to define reusable lists of objects for other utilities to process. You can use TEMPLATE to define templates for data sets that are allocated by other utilities. LISTDEF and TEMPLATE are often used together.

LISTDEF

When you define a list with the LISTDEF utility, you can specify the objects that you want to include in the list by using either explicit names or naming patterns with wildcard characters. You can then specify this list by name in other utility control statements. Using such predefined lists saves you from having to specify all of these objects each time you run a utility or having to run the utility multiple times. These lists can also help ensure that an object is not accidentally omitted. For example, if you want to make an image copy of every table space in database ABC, you can specify such a list (`LISTDEF ABCLIST INCLUDE TABLESPACES DATABASE ABC`) and then run the COPY utility on this list. Utility processing builds the list of objects during execution.

Tip: Colloquially, the lists that are defined by the LISTDEF utility are also referred to as *LISTDEFs*, or list definitions. So be aware that *LISTDEF* can either refer to the utility or the list itself.

Related information:

Syntax and options of the LISTDEF control statement (Db2 12 for z/OS)
"Running utilities on LISTDEF lists" on page 609

TEMPLATE

When you define a data set template by using the TEMPLATE utility, you specify the data set naming convention as well as other data set attributes. You can then specify this template in other utility statements instead of DD names. Those utilities will use the template specifications when allocating the data set. The data set name is constructed during utility processing based on the data set name expression that is specified in the TEMPLATE utility control statement. That expression often includes an object-identifying pattern, such as database or space name, and other variables. These patterns allows a single utility job step to include many objects.

Using templates eliminates the need for certain JCL DD statements during utility processing. Templates can also help standardize data set names.

You can use templates together with lists that were defined by LISTDEF. For example, a single utility can process many objects from a LISTDEF list and then dynamically define the target output data sets by using templates. However, you do not have to use templates with such lists; a utility that is not processing a list can still use one or more templates.

Db2 Admin Tool also supports the use of templates for Db2 Admin Tool work data sets that are created and used in the jobs that are generated for the following functions: alter, restore, redefine, migrate, and object comparison. As with the utility data sets, templates allow you to define your own data set naming convention and also control other allocation attributes for the data sets. The set of variables that can be specified for the data set names for these non-utility work data sets depends on the Db2 Admin Tool function.

Important: Both Db2 Admin Tool and Db2 Object Comparison Tool support the use of REORG and COPY utilities in the Alter, OC, and CM functions. If COPYDDN 1 and COPYDDN 2 templates are specified, specify a unique symbolic variable to prevent conflicts.

Related information:

Syntax and options of the TEMPLATE control statement (Db2 12 for z/OS)
"User-defined or product default templates" on page 444

Creating the LISTDEF control tables

Before you can create and use LISTDEFs, you must create two Db2 control tables to store the LISTDEF definitions.

About this task

These tables have the following default names:

DSNACC.UTLIST
Contains basic LISTDEF definitions.

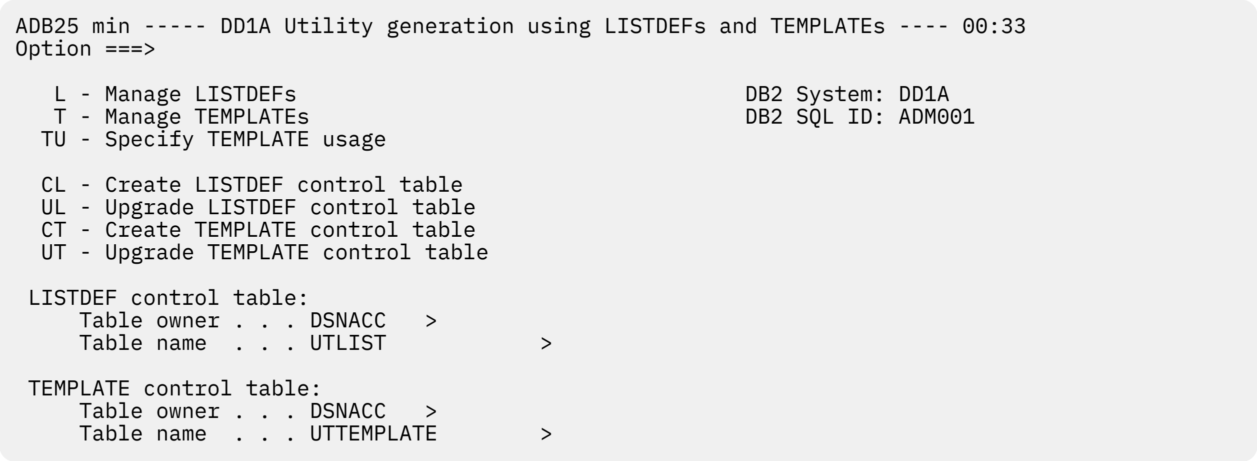
DSNACC.UTLISTE
Contains detailed LISTDEF definitions.

Before proceeding with the following steps, determine whether these tables already exist. If they do exist, go to "Editing a LISTDEF" on page 421.

Procedure

1. On the **Administration Menu** panel, specify option 5, and press Enter.

   The **Utility generation using LISTDEFs and TEMPLATEs** panel is displayed, as shown in the following figure.

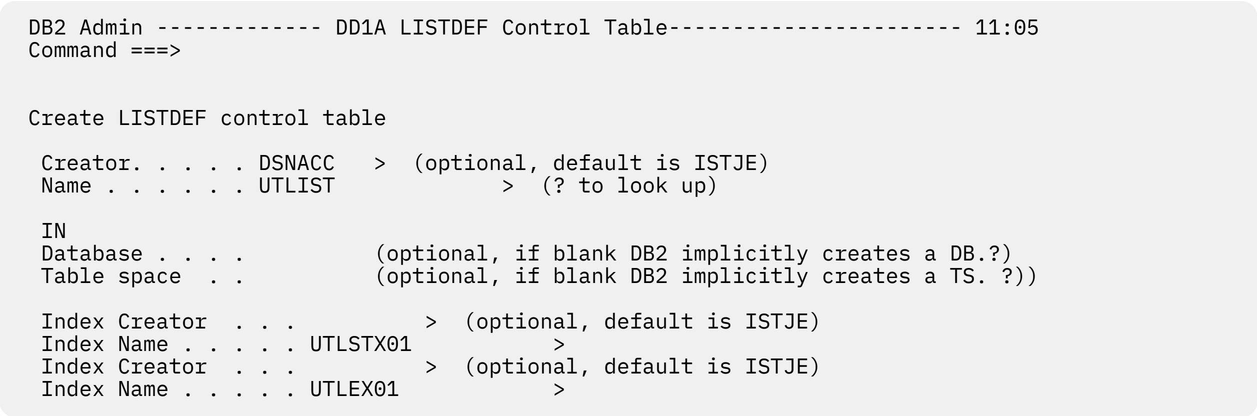


```
ADB25 min ----- DD1A Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

   L - Manage LISTDEFs                                DB2 System: DD1A
   T - Manage TEMPLATEs                               DB2 SQL ID: ADM001
  TU - Specify TEMPLATE usage

  CL - Create LISTDEF control table
  UL - Upgrade LISTDEF control table
  CT - Create TEMPLATE control table
  UT - Upgrade TEMPLATE control table

 LISTDEF control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTLIST            >

 TEMPLATE control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTTEMPLATE        >
```

Figure 209. Utility generation using LISTDEFs and TEMPLATEs panel (ADB25)

2. Specify option CL, and press Enter.

   The **LISTDEF Control Table** panel is displayed, as shown in the following figure.



```
DB2 Admin ------------- DD1A LISTDEF Control Table----------------------- 11:05
Command ===>


Create LISTDEF control table

 Creator. . . . . DSNACC   >  (optional, default is ISTJE)
 Name . . . . . . UTLIST             >  (? to look up)

 IN
 Database . . . .          (optional, if blank DB2 implicitly creates a DB.?)
 Table space  . .          (optional, if blank DB2 implicitly creates a TS. ?))

 Index Creator  . . .          >  (optional, default is ISTJE)
 Index Name . . . . . UTLSTX01          >
 Index Creator  . . .          >  (optional, default is ISTJE)
 Index Name . . . . . UTLEX01           >
```

Figure 210. LISTDEF/TEMPLATE Control Table panel (ADB25C)

3. Specify values or accept the default values for the following fields:

Creator
The creator of the control table.

Name
The name of the control table. The control table that contains detailed LISTDEF definitions is automatically appended with the letter E.

Database
The name of the database.

Table Space
The name of the table space.

Basic LISTDEF definition table information
The first set of **Index Creator** and **Index Name** fields are used for the basic LISTDEF definition table.

Index Creator
The name of the index creator.

Index name
The name of the index.

The default value is DSNACC.UTLIST.

Detailed LISTDEF definition table information
The second set of **Index Creator** and **Index Name** fields are used for the detailed LISTDEF definition table.

Index Creator
The name of the index creator.

Index name
The name of the index.

The default value is DSNACC.UTLISTE.

4. Press Enter to create the tables.

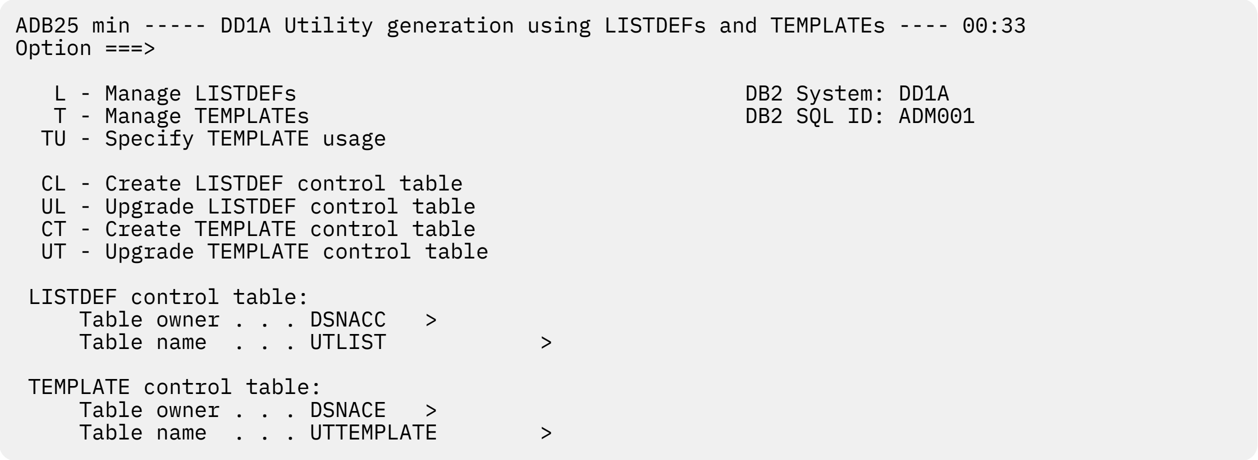
Upgrading the LISTDEF control tables

Use the UL command option to upgrade a LISTDEF control table to the current Db2 version.

Procedure

1. On the **Administration Menu** panel, specify option 5, and press Enter.

   The **Utility generation using LISTDEFs and TEMPLATEs** panel is displayed, as shown in the following figure.



```
ADB25 min ----- DD1A Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

   L - Manage LISTDEFs                                 DB2 System: DD1A
   T - Manage TEMPLATEs                                DB2 SQL ID: ADM001
  TU - Specify TEMPLATE usage

  CL - Create LISTDEF control table
  UL - Upgrade LISTDEF control table
  CT - Create TEMPLATE control table
  UT - Upgrade TEMPLATE control table

 LISTDEF control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTLIST            >

 TEMPLATE control table:
     Table owner . . . DSNACE   >
     Table name  . . . UTTEMPLATE        >
```

Figure 211. Utility generation using LISTDEFs and TEMPLATEs panel (ADB25)

2. On the Option command line, specify option UL, and press Enter.

   The table is validated to make sure it is a LISTDEF control table. The validation is based on the following column names and data types: NAME VARCHAR(18), TYPE VARCHAR(2), CREATEDBY VARCHAR(8), MODIFIEDBY VARCHAR(8), REMARKS VARCHAR(254). If the LISTDEF control table name is not at the current version, an upgrade is performed.

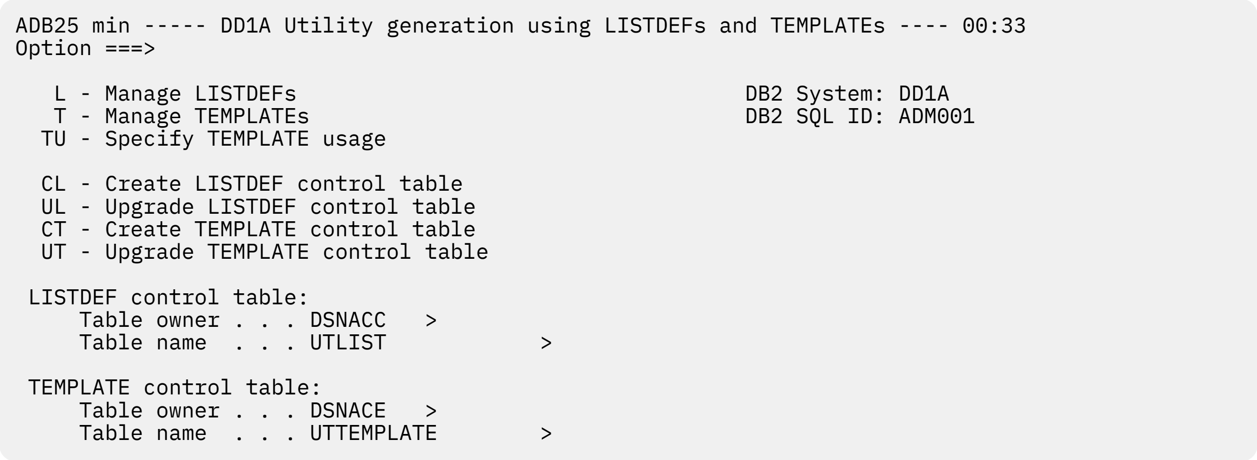
Adding a LISTDEF

Use the **LISTDEFs** panel to add a LISTDEF to the LISTDEF control tables.

Procedure

1. On the **Administration Menu** panel, specify option 5, and press Enter.

   The **Utility generation using LISTDEFs and TEMPLATEs** panel is displayed, as shown in the following figure.

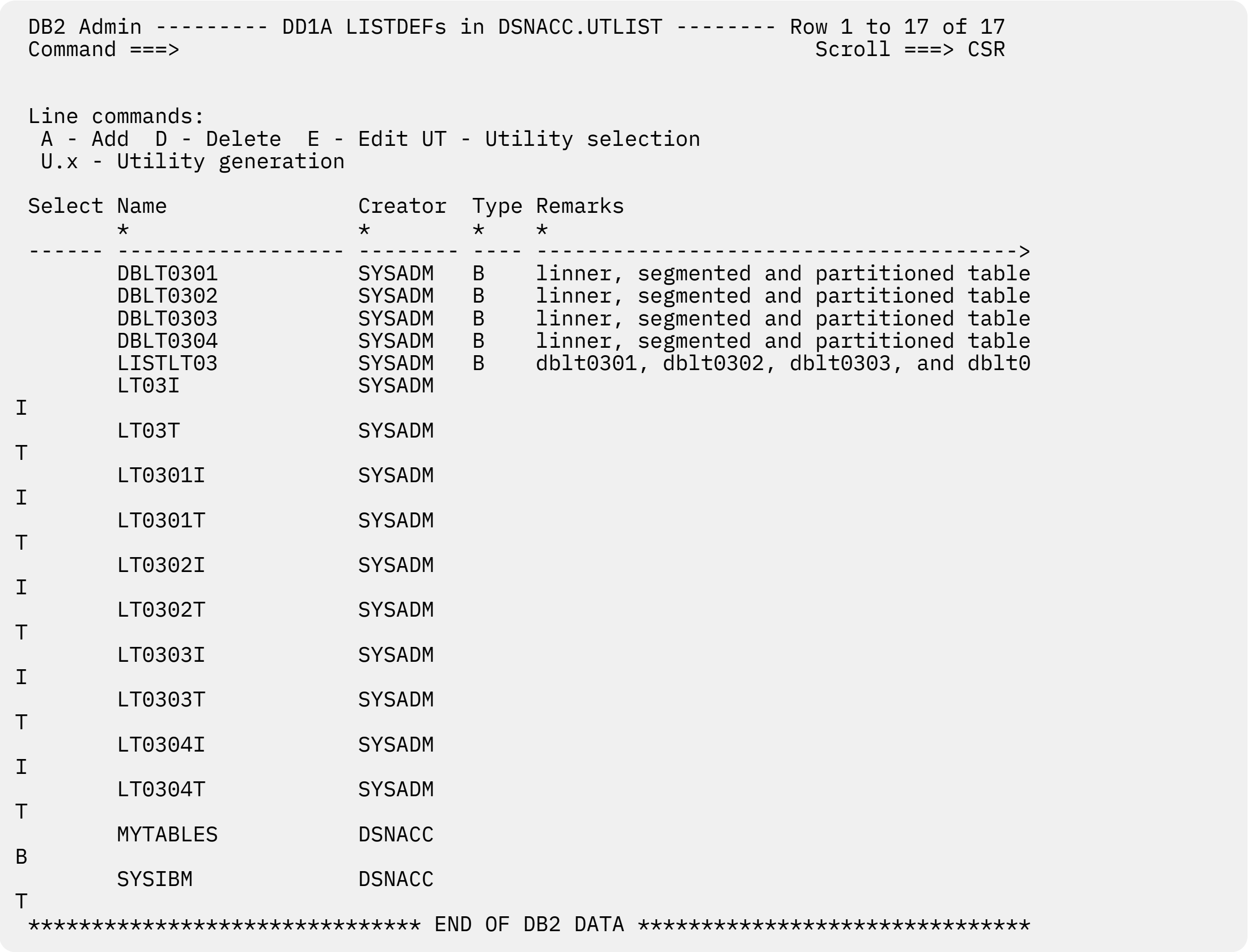


```
ADB25 min ----- DD1A Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

   L - Manage LISTDEFs                                   DB2 System: DD1A
   T - Manage TEMPLATEs                                  DB2 SQL ID: ADM001
  TU - Specify TEMPLATE usage

  CL - Create LISTDEF control table
  UL - Upgrade LISTDEF control table
  CT - Create TEMPLATE control table
  UT - Upgrade TEMPLATE control table

 LISTDEF control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTLIST            >

 TEMPLATE control table:
     Table owner . . . DSNACE   >
     Table name  . . . UTTEMPLATE        >
```

Figure 212. Utility generation using LISTDEFs and TEMPLATEs panel (ADB25)

2. Specify option L, and press Enter.

   The **LISTDEFs** panel is displayed, as shown in the following figure.



```
 DB2 Admin --------- DD1A LISTDEFs in DSNACC.UTLIST -------- Row 1 to 17 of 17
 Command ===>                                                  Scroll ===> CSR


 Line commands:
  A - Add  D - Delete  E - Edit UT - Utility selection
  U.x - Utility generation

 Select Name               Creator  Type Remarks
        *                  *        *    *
 ------ ----------------- -------- ---- ------------------------------------->
        DBLT0301           SYSADM   B    linner, segmented and partitioned table
        DBLT0302           SYSADM   B    linner, segmented and partitioned table
        DBLT0303           SYSADM   B    linner, segmented and partitioned table
        DBLT0304           SYSADM   B    linner, segmented and partitioned table
        LISTLT03           SYSADM   B    dblt0301, dblt0302, dblt0303, and dblt0
        LT03I              SYSADM
I
        LT03T              SYSADM
T
        LT0301I            SYSADM
I
        LT0301T            SYSADM
T
        LT0302I            SYSADM
I
        LT0302T            SYSADM
T
        LT0303I            SYSADM
I
        LT0303T            SYSADM
T
        LT0304I            SYSADM
I
        LT0304T            SYSADM
T
        MYTABLES           DSNACC
B
        SYSIBM             DSNACC
T
 ******************************* END OF DB2 DATA *******************************
```

Figure 213. LISTDEFs panel (ADB25L)

The following fields are on this panel:

SEL
Input field where you enter the line command. The following line commands are valid:

A
Adds a new LISTDEF.

D
Deletes a LISTDEF.

E
Edits a LISTDEF.

UT
Invokes a utility against a LISTDEF.

U.x
Generates a utility job stream. Substitute 'x' with the LISTDEF utility option. For example, U.TU specifies use of a template for utility JCL and work statement list output.

NAME
The name of a LISTDEF.

CREATOR
Creator of the definition or the last ID to update it.

TYPE
This field can have one of three values. For LISTDEFs that are added with Db2 Admin Tool, the value is B, which is the default value. This field is updatable.

T
Table space.

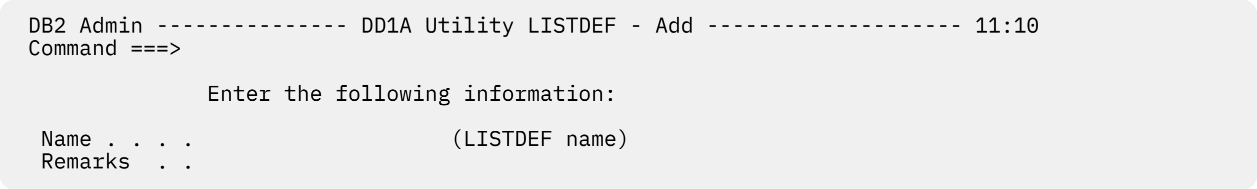
I
Index space.

B
Both table space and index space.

REMARKS
This field contains an optional description of the LISTDEF. You can modify this field.

3. Issue the A line command, and press Enter.

   The **Add LISTDEF** panel is displayed, shown in the following figure.



```
DB2 Admin --------------- DD1A Utility LISTDEF - Add -------------------- 11:10
Command ===>

              Enter the following information:

 Name . . . .                    (LISTDEF name)
 Remarks  . .
```

Figure 214. Add LISTDEFs panel (ADB25LA)

   The following fields are on the panel:

   NAME
   The name of the LISTDEF. This name must be unique for the control table being used.

   REMARKS
   An optional description of the LISTDEF.

4. Enter a unique name for the LISTDEF, identify the type of objects that the LISTDEF will apply to (T for table spaces, I for index spaces, or B for both) and optionally include a description of the LISTDEF.
5. Press Enter to add the LISTDEF to the LISTDEF control tables.

Editing a LISTDEF

With Db2 Admin Tool, you can add, delete, or edit a clause contained in a LISTDEF.

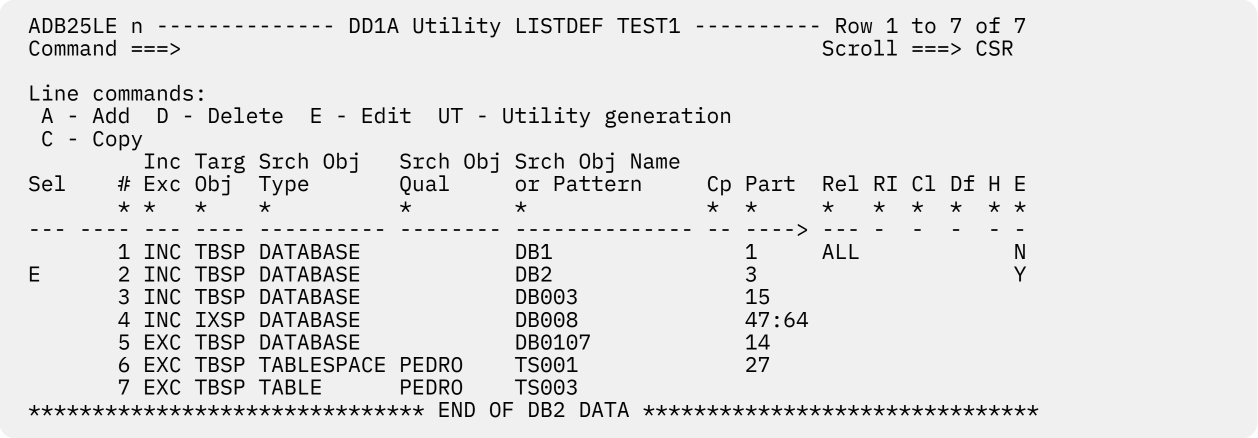
About this task

Each LISTDEF consists of one or more clauses; each clause represents a separate line on the panel. When you initially define a LISTDEF, an empty clause is created. Use the following instructions to complete the definition of a new, empty clause, to edit an existing clause, or to delete a clause. You then fill in the fields to complete the definition of the clause; if you fail to fill in a required field, Db2 Admin Tool prompts you for it. After a clause is created, you can edit it by typing over the field you wish to change or you can enter an E to the left of the clause to be changed. This latter approach can be used to edit a single clause.

Procedure

1. From the **LISTDEFs** panel, issue the E line command against the LISTDEF that you want to edit, and press Enter.

   The **Edit LISTDEF** panel is displayed, as shown in the following figure.



```
ADB25LE n -------------- DD1A Utility LISTDEF TEST1 ---------- Row 1 to 7 of 7
Command ===>                                                  Scroll ===> CSR

Line commands:
 A - Add  D - Delete  E - Edit  UT - Utility generation
 C - Copy
         Inc Targ Srch Obj   Srch Obj Srch Obj Name
Sel    # Exc Obj  Type       Qual     or Pattern     Cp Part  Rel RI Cl Df H E
       * *   *    *          *        *              *  *     *   *  *  *  * *
--- ---- --- ---- ---------- -------- -------------- -- ----> --- -  -  -  - -
       1 INC TBSP DATABASE            DB1              1     ALL            N
E      2 INC TBSP DATABASE            DB2              3                    Y
       3 INC TBSP DATABASE            DB003            15
       4 INC IXSP DATABASE            DB008            47:64
       5 EXC TBSP DATABASE            DB0107           14
       6 EXC TBSP TABLESPACE PEDRO    TS001            27
       7 EXC TBSP TABLE      PEDRO    TS003
****************************** END OF DB2 DATA ******************************
```

Figure 215. Edit LISTDEF control table panel (ADB25LE)

New, empty clauses are identified by a question mark (?) in the **Incl Excl** column.

The following fields are on this panel:

SEL
Action field where you enter the line command. The following line commands are valid:

A
Adds a new clause to the LISTDEF.

D
Deletes a clause.

E
Edits a LISTDEF clause. Use the **Edit LISTDEF clause** panel to edit a single clause.

UT
Invokes a utility against a single clause of the LISTDEF.

C
Creates a copy of the selected clause.

#
The sequence number is part of a unique key which means that no two clauses within the same LISTDEF can have the same sequence number. The sequence of your clauses is important, because clauses are executed in ascending order. If you need to reorder the clauses in a LISTDEF, make room by updating the lowest clause that needs to be changed with a sequence number greater than the others, then renumbering the rest as needed.

INC EXC
Include or exclude objects based on the search criteria. It is sufficient to enter I or E.

TARG OBJ
This field refers to whether a list of table spaces or index spaces is to be created. It is sufficient to enter T for table spaces or I for index spaces.

SRCH OBJ TYPE
This field refers to the type of object for which to search. The following values are permissible:

D
Database

L
List

T
Table

TS
Table space

I or IX

Index

IS

Index space

SRCH OBJ QUAL

For object types table and index, this field indicates the owner.

For object types table space and index space, this field indicates the database name.

For certain object types, partial or complete wild-carding is available by using an asterisk (*). For example, `DB01*`.

SRCH OBJ NAME OR PATTERN

This field indicates the name of the search object, with partial or complete wild-carding available for certain object types. The wild card character is the asterisk (*).

CP

This field refers to COPY YES or COPY NO, and is applicable only to index spaces. For COPY YES, enter `Y`. For COPY NO, enter `N`.

Part

This field refers to the PARTLEVEL keyword, or, if a number is specified, to the partition that is to be included. Permissible values are:

blank

The PARTLEVEL keyword is not added to the LISTDEF clause. As a result, the entire set of partitions in a partitioned table space is included as one unit. A sample LISTDEF might look like this:

```
LISTDEF T -- 00000010 OBJECTS
  INCLUDE TABLESPACE R148286.DB2CLEAN
  INCLUDE TABLESPACE R148286.DSN8S81D
  INCLUDE TABLESPACE R148286.DSN8S81E
  INCLUDE TABLESPACE R148286.DSN8S81P
  INCLUDE TABLESPACE R148286.EMP1
  INCLUDE TABLESPACE R148286.PART
  INCLUDE TABLESPACE R148286.PLANRTAB
  INCLUDE TABLESPACE R148286.T1
  INCLUDE TABLESPACE R148286.T2
  INCLUDE TABLESPACE R148286.T3
```

Y

Each partition is included as a separate object; the result might look like this:

```
LISTDEF T -- 00000014 OBJECTS
  INCLUDE TABLESPACE R148286.DB2CLEAN
  INCLUDE TABLESPACE R148286.DSN8S81D
  INCLUDE TABLESPACE R148286.DSN8S81E PARTLEVEL(00001)
  INCLUDE TABLESPACE R148286.DSN8S81E PARTLEVEL(00002)
  INCLUDE TABLESPACE R148286.DSN8S81E PARTLEVEL(00003)
  INCLUDE TABLESPACE R148286.DSN8S81E PARTLEVEL(00004)
  INCLUDE TABLESPACE R148286.DSN8S81P
  INCLUDE TABLESPACE R148286.EMP1
  INCLUDE TABLESPACE R148286.PART PARTLEVEL(00001)
  INCLUDE TABLESPACE R148286.PART PARTLEVEL(00002)
  INCLUDE TABLESPACE R148286.PLANRTAB
  INCLUDE TABLESPACE R148286.T1
  INCLUDE TABLESPACE R148286.T2
  INCLUDE TABLESPACE R148286.T3
```

1-4096

Enter a single partition number in this range for it to be included. The resultant LISTDEF might look like the following example:

```
LISTDEF T -- 00000010 OBJECTS
  INCLUDE TABLESPACE R148286.DB2CLEAN
  INCLUDE TABLESPACE R148286.DSN8S81D
  INCLUDE TABLESPACE R148286.DSN8S81E PARTLEVEL(00001)
  INCLUDE TABLESPACE R148286.DSN8S81P
  INCLUDE TABLESPACE R148286.EMP1
```

```
   INCLUDE TABLESPACE R148286.PART PARTLEVEL(00001)
   INCLUDE TABLESPACE R148286.PLANRTAB
   INCLUDE TABLESPACE R148286.T1
   INCLUDE TABLESPACE R148286.T2
   INCLUDE TABLESPACE R148286.T3
```

integer1:integer2
Starting with DB2 10 for z/OS, the partitions can be specified as a range. integer1:integer2 indicates the range of partitions to be specified in a list.

Rel
Auxiliary relationship can be ALL, BASE, LOB or XML. Specify one of the following values:

A
Enter an A for ALL (base table spaces, related index spaces, and large objects).

B
Enter a B for base table spaces and related index spaces.

L
Enter an L for a large object.

X
Enter an X for an XML object.

RI
Specify Y to include objects that are related through referential integrity.

CI
Filter the objects returned based on the existence or absence of cloned objects. The value can by Y or N

Df
Filter the LISTDEF objects based on whether data sets are defined or not. The value can be Y, N, A (all)

H
Specifies that only history objects should be included in the results.

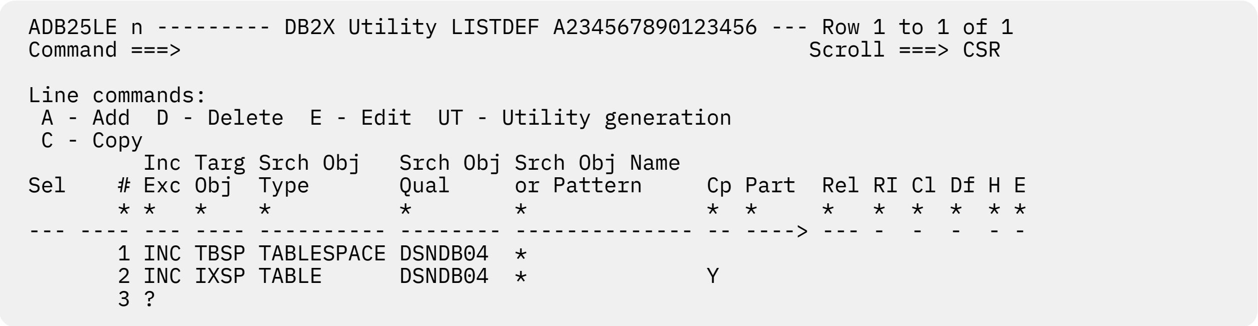
E
Filter the objects returned by the LISTDEF based on the format of the RBA or LRSN.

Y
Only objects with extended format are selected.

N
Only objects with basic format are selected.

2. To edit existing clauses, you can either type over the field or fields that you want to change or you can issue the E line command to edit a single clause.
3. To add a clause, issue the A line command.

   A new empty clause, as identified by a question mark (?), is inserted, as shown in the panel in the following figure.



```
ADB25LE n --------- DB2X Utility LISTDEF A234567890123456 --- Row 1 to 1 of 1
Command ===>                                                 Scroll ===> CSR

Line commands:
 A - Add  D - Delete  E - Edit  UT - Utility generation
 C - Copy
         Inc Targ Srch Obj   Srch Obj Srch Obj Name
Sel    # Exc Obj  Type       Qual     or Pattern     Cp Part  Rel RI Cl Df H E
       * *   *    *          *        *              *  *     *   *  *  *  * *
--- ---- --- ---- ---------- -------- ------------- -- ----> --- -  -  -  - -
       1 INC TBSP TABLESPACE DSNDB04  *
       2 INC IXSP TABLE      DSNDB04  *              Y
       3 ?
```

Figure 216. LISTDEF panel (ADB25LE) – adding a clause

4. Type in the fields to complete the definition of the clause and press Enter to complete the addition.

Alternatively, you can enter an E to the left of the clause to bring up the **Edit LISTDEF clause** panel, which can be used to edit a single clause.

5. To delete a clause, issue the D line command against the clause that you want to delete.

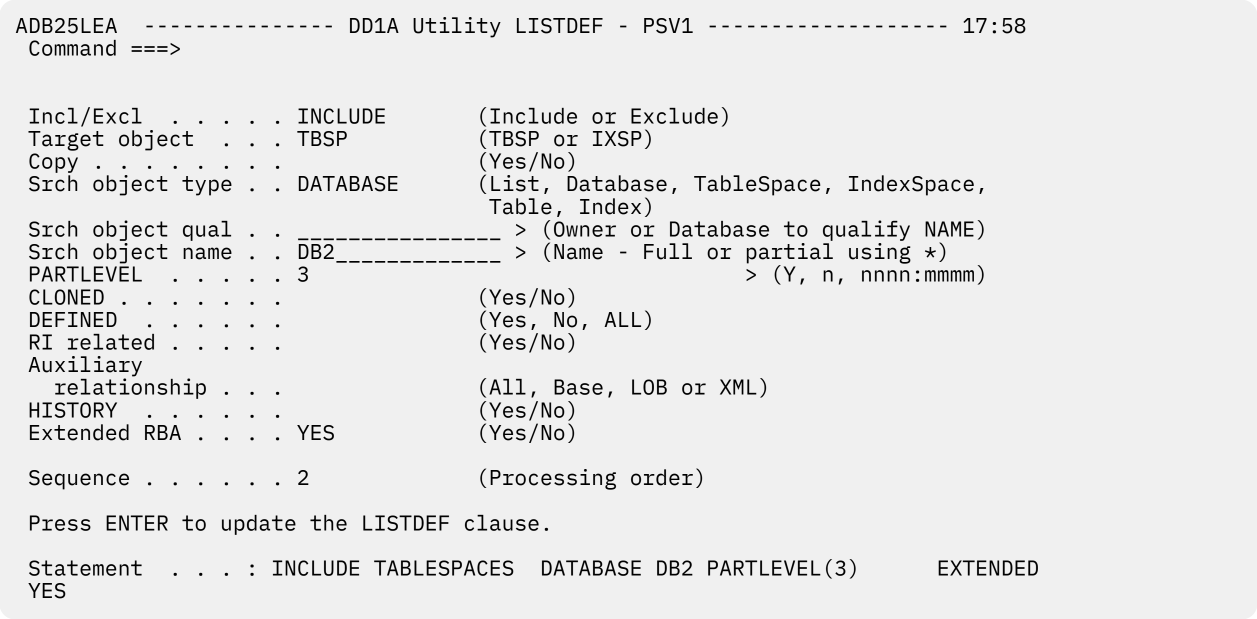
Editing a single LISTDEF clause

Use the **Edit LISTDEF clause** panel to edit a single LISTDEF clause.

Procedure

1. To display the Edit LISTDEF clause, issue the E line command against a LISTDEF.

   The following figure shows the **Edit LISTDEF clause** panel.



```
ADB25LEA  --------------- DD1A Utility LISTDEF - PSV1 ------------------- 17:58
 Command ===>


 Incl/Excl  . . . . . INCLUDE       (Include or Exclude)
 Target object  . . . TBSP          (TBSP or IXSP)
 Copy . . . . . . . .               (Yes/No)
 Srch object type . . DATABASE      (List, Database, TableSpace, IndexSpace,
                                     Table, Index)
 Srch object qual . . ________________ > (Owner or Database to qualify NAME)
 Srch object name . . DB2_____________ > (Name - Full or partial using *)
 PARTLEVEL  . . . . . 3                                > (Y, n, nnnn:mmmm)
 CLONED . . . . . . .               (Yes/No)
 DEFINED  . . . . . .               (Yes, No, ALL)
 RI related . . . . .               (Yes/No)
 Auxiliary
   relationship . . .               (All, Base, LOB or XML)
 HISTORY  . . . . . .               (Yes/No)
 Extended RBA . . . . YES           (Yes/No)

 Sequence . . . . . . 2             (Processing order)

 Press ENTER to update the LISTDEF clause.

 Statement  . . . : INCLUDE TABLESPACES  DATABASE DB2 PARTLEVEL(3)      EXTENDED
 YES
```

Figure 217. Edit LISTDEF clause panel (ADB25LEA)

2. Specify the following values.

   As you enter information in the fields, the generated LISTDEF clause is shown at the bottom of the panel.

   The following fields are shown on this panel:

   INCL/EXCL
   Include or exclude objects based on the search criteria. It is sufficient to enter I include objects or E to exclude objects.

   TARGET OBJ
   Permissible values are:

   T
   Table space

   I
   Index space

   COPY
   This field refers to COPY YES or COPY NO, and is applicable only to index spaces. For COPY YES, enter Y. For COPY NO, enter N.

   SRCH OBJ TYPE
   This field refers to the type of Db2 Admin Tool Look Up object for the initial search. The following values are permissible:

L
List

D
Database

TS
Table space

IS
Index space

TB
Table

I or IX
Index

SRCH OBJ QUAL
For Db2 Admin Tool Look Up types table and index, this field indicates the owner.

For Db2 Admin Tool Look Up types table space and index space, this field indicates the database name.

For some Db2 Admin Tool Look Up types, partial or complete wild-carding is available by using an asterisk (*). For example, DB01*.

SRCH OBJ NAME
This field indicates the name of the Db2 Admin Tool Look Up object, with partial or complete wild-carding available for some Db2 Admin Tool Look Up types. The wild card character is the asterisk (*).

PARTLEVEL
This field refers to the PARTLEVEL keyword, or, if a number is specified, to the partition that is to be included.

RI related
Specify Y to include objects that are related through referential integrity.

HISTORY
A filtering keyword that specifies that only history (versioning) objects should be included on the resulting list clause.

Extended RBA
Filter the objects returned by the LISTDEF based on the format of the RBA or LRSN.

- Yes - only objects with extended format are selected.
- No - only objects with basic format are selected.

Auxiliary relationship
This field indicates a large object type. Specify one of the following values:

A
Specify an A for ALL (base table spaces, related index spaces, and large objects).

B
Specify a B for base table spaces and related index spaces.

L
Specify an L for LOB.

X
Specify an X for XML.

Sequence
The sequence number is part of a unique key, which means that no two clauses with the same LISTDEF can have the same sequence number. The sequence of your clauses is important because they are executed in ascending order. If you need to reorder the clauses in a LISTDEF, create room

by updating the lowest clause that needs to be changed with a sequence number greater than the others; then renumber the rest as needed.

Deleting a LISTDEF

Use the **LISTDEFs** panel to delete a LISTDEF from the LISTDEF control tables.

Procedure

1. On the **Administration Menu** panel, specify option 5, and press Enter.

   The **Utility generation using LISTDEFs and TEMPLATEs** panel is displayed.
2. Specify option L, and press Enter.

   The **LISTDEFs** panel is displayed.
3. Issue the D line command, and press Enter to delete the corresponding LISTDEF from the LISTDEF control tables.

Results

The LISTDEF is removed from the control tables.

Adding templates

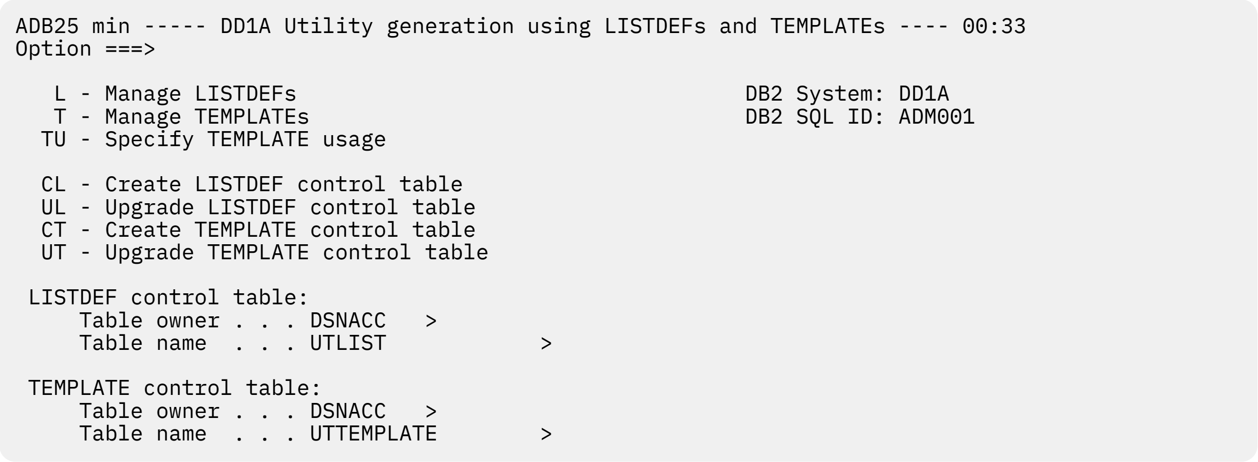
You can create data set templates that are defined by the TEMPLATE utility. These templates can be used by other utilities when allocating data sets instead of JCL DD statements.

Procedure

To add a template:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 5, and press Enter.

   The **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel is displayed.

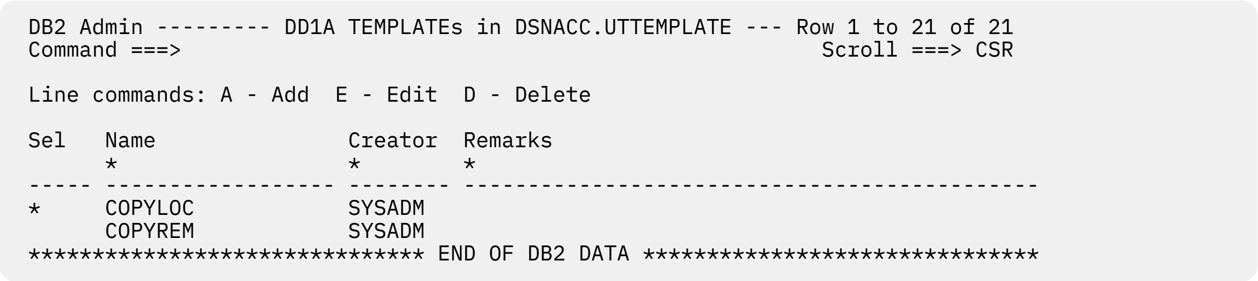


```
ADB25 min ----- DD1A Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

   L - Manage LISTDEFs                                   DB2 System: DD1A
   T - Manage TEMPLATEs                                  DB2 SQL ID: ADM001
  TU - Specify TEMPLATE usage

  CL - Create LISTDEF control table
  UL - Upgrade LISTDEF control table
  CT - Create TEMPLATE control table
  UT - Upgrade TEMPLATE control table

 LISTDEF control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTLIST            >

 TEMPLATE control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTTEMPLATE        >
```

*Figure 218. **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel*

2. Specify option T, and press Enter.

   The **TEMPLATEs (ADB25T)** panel is displayed, as shown in the following figure:



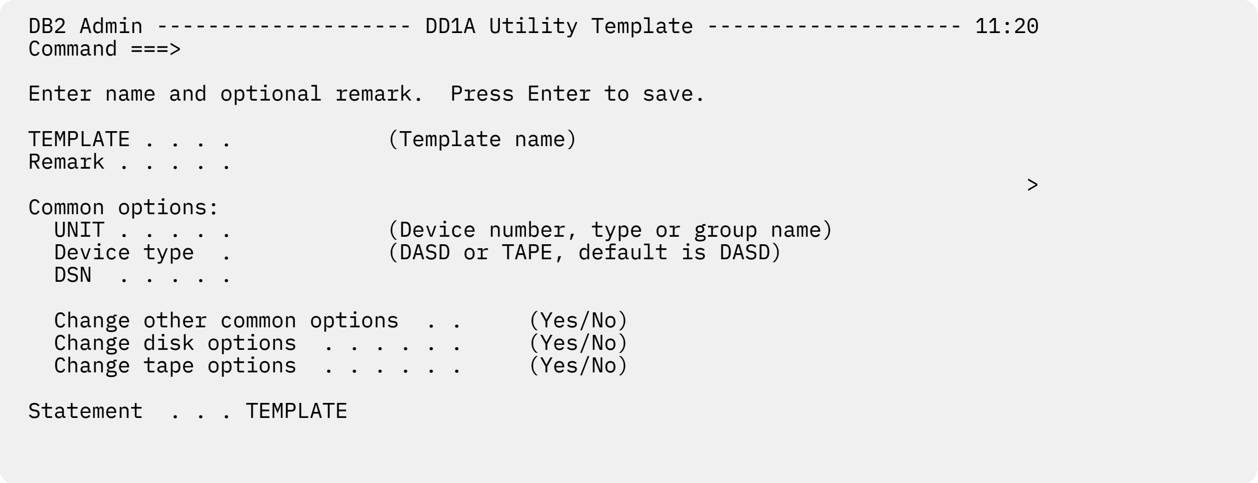
```
DB2 Admin --------- DD1A TEMPLATEs in DSNACC.UTTEMPLATE --- Row 1 to 21 of 21
Command ===>                                                  Scroll ===> CSR

Line commands: A - Add  E - Edit  D - Delete

Sel   Name               Creator  Remarks
      *                  *        *
----- ------------------ -------- ---------------------------------------------
*     COPYLOC            SYSADM
      COPYREM            SYSADM
******************************* END OF DB2 DATA *******************************
```

Figure 219. ***TEMPLATEs (ADB25T)*** *panel*

It shows the existing templates within the control table. The table name is in the panel header. For example, in the previous figure, the table name is DSNACC.UTTEMPLATE.

3. Specify option A in the Sel column of the first row, and press Enter.

   The **Utility Template (ADB25TE)** panel is displayed, as shown in the following figure:



```
DB2 Admin -------------------- DD1A Utility Template -------------------- 11:20
Command ===>

Enter name and optional remark.  Press Enter to save.

TEMPLATE . . . .            (Template name)
Remark . . . . .
                                                                              >
Common options:
  UNIT . . . . .            (Device number, type or group name)
  Device type  .            (DASD or TAPE, default is DASD)
  DSN  . . . . .

  Change other common options  . .     (Yes/No)
  Change disk options  . . . . . .     (Yes/No)
  Change tape options  . . . . . .     (Yes/No)

Statement  . . . TEMPLATE
```

Figure 220. Utility Template panel (ADB25TE)

4. Specify values in the following fields:

   Required fields

   TEMPLATE

   A name for the template. The name must be unique within the control table that you are using.

   DSN

   A data set name pattern for the template. The data set name can be composed of variables whose value is determined and substituted during execution of the utility that is using the template or execution of the job that Db2 Admin Tool generated for alter, restore, redefine, migrate, or object comparison processing that is using the template.

   To construct a data set name pattern by using substitution variables, specify a question mark (?) as the first character of the **DSN** field. When you press Enter, the **Utility Template – Data Set Name (ADB25TD)** panel is displayed. For more information about data set names and substitution variables, see "Creating a template DSN by selecting substitution variables" on page 435

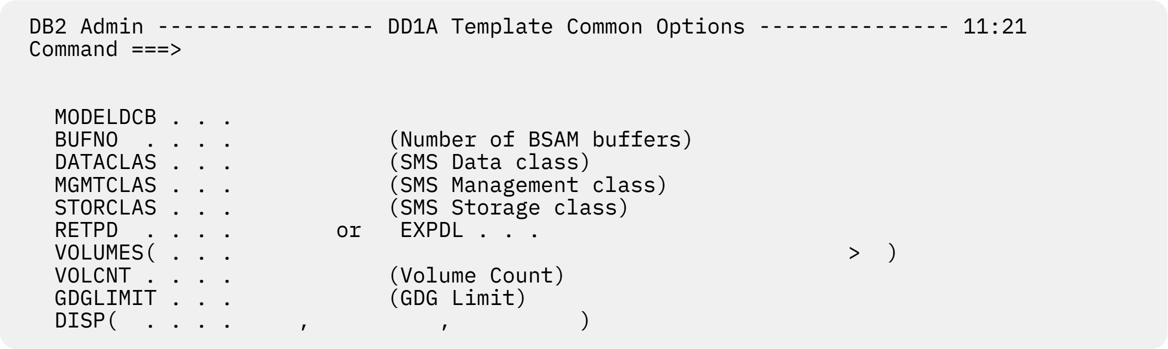
   The variables displayed on the **Utility Template – Data Set Name (ADB25TD)** panel are the variables that are supported for normal Db2 utility template processing. Therefore, any variable displayed is valid for the data set name pattern for a utility data set template. However, not all of the variables are valid for the templates for non-utility work data sets, and additional variables might apply.

Optional fields

UNIT
The device number or group name for the data set.

Device type
The device type for the data set.

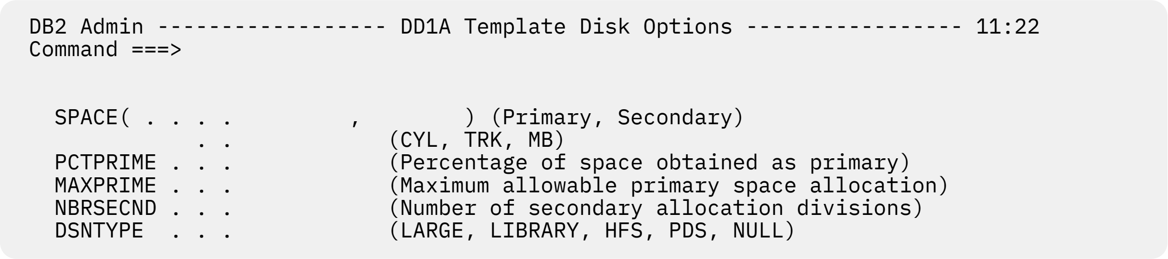
Change other common options
Additional attributes for the data set. When you specify Yes and press Enter, the **Template Common Options (ADB25TC)** panel is displayed, as shown in the following figure. See the online help for the description of the fields on this panel.



```
DB2 Admin ----------------- DD1A Template Common Options --------------- 11:21
Command ===>


  MODELDCB . . .
  BUFNO  . . . .            (Number of BSAM buffers)
  DATACLAS . . .            (SMS Data class)
  MGMTCLAS . . .            (SMS Management class)
  STORCLAS . . .            (SMS Storage class)
  RETPD  . . . .        or   EXPDL . . .
  VOLUMES( . . .                                                 >  )
  VOLCNT . . . .            (Volume Count)
  GDGLIMIT . . .            (GDG Limit)
  DISP(  . . . .     ,          ,          )
```

*Figure 221. **Template Common Options (ADB25TC)** panel*

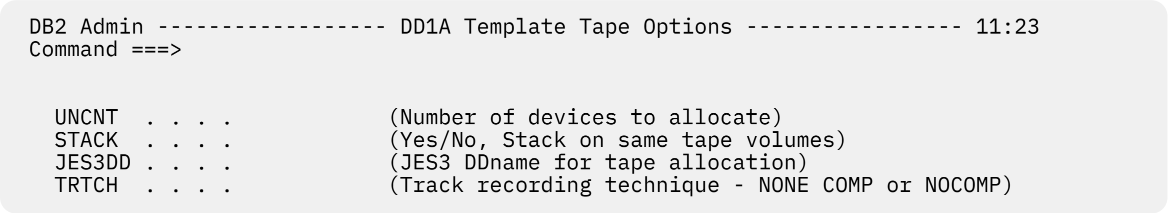
Change disk options
Additional options for the data set that are applicable only to data sets that are on disk. When you specify Yes and press Enter, the **Template Disk Options (ADB25TS)** panel is displayed, as shown in the following figure. See the online help for the description of the fields on the panel.



```
DB2 Admin ------------------ DD1A Template Disk Options ----------------- 11:22
Command ===>


  SPACE( . . . .         ,        ) (Primary, Secondary)
             . .            (CYL, TRK, MB)
  PCTPRIME . . .            (Percentage of space obtained as primary)
  MAXPRIME . . .            (Maximum allowable primary space allocation)
  NBRSECND . . .            (Number of secondary allocation divisions)
  DSNTYPE  . . .            (LARGE, LIBRARY, HFS, PDS, NULL)
```

*Figure 222. **Template Disk Options (ADB25TS)** panel*

Change tape options
Additional options for the data set that are applicable only for data sets on tape. When you specify Yes and press Enter, the **Template Tape Options (ADB25TT)** panel is displayed, as shown in the following figure. See the online help for the description of the fields on the panel.



```
DB2 Admin ------------------ DD1A Template Tape Options ----------------- 11:23
Command ===>


  UNCNT  . . . .            (Number of devices to allocate)
  STACK  . . . .            (Yes/No, Stack on same tape volumes)
  JES3DD . . . .            (JES3 DDname for tape allocation)
  TRTCH  . . . .            (Track recording technique - NONE COMP or NOCOMP)
```

*Figure 223. **Template Tape Options (ADB25TT)** panel*

5. Press Enter.

   The template is added.

Editing templates

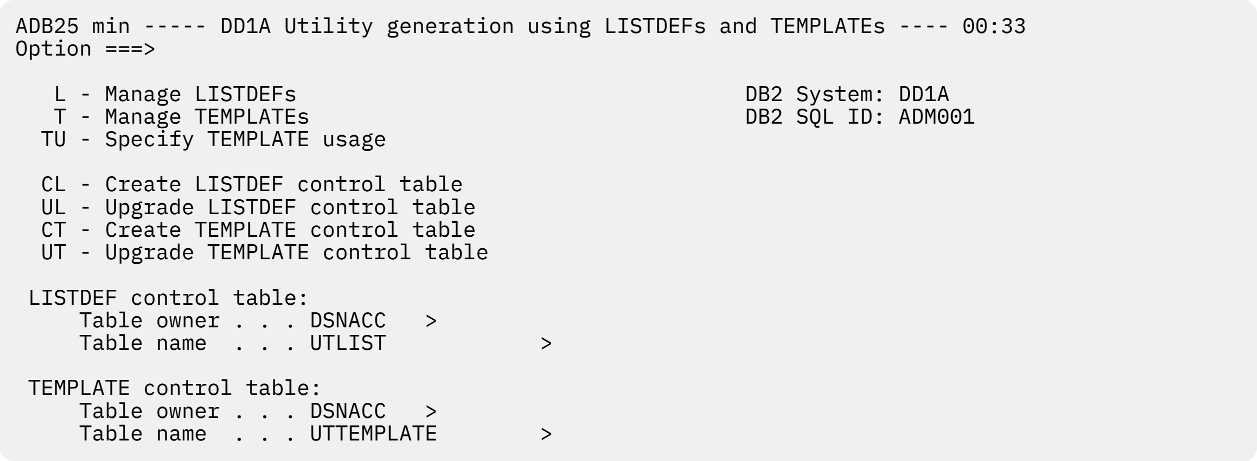
You can edit existing data set templates that are defined by the TEMPLATE utility. For example, you can change the pattern for the data set name, the device type, the unit, and other data set attributes.

Procedure

To edit a template:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 5, and press Enter.

   The **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel is displayed.

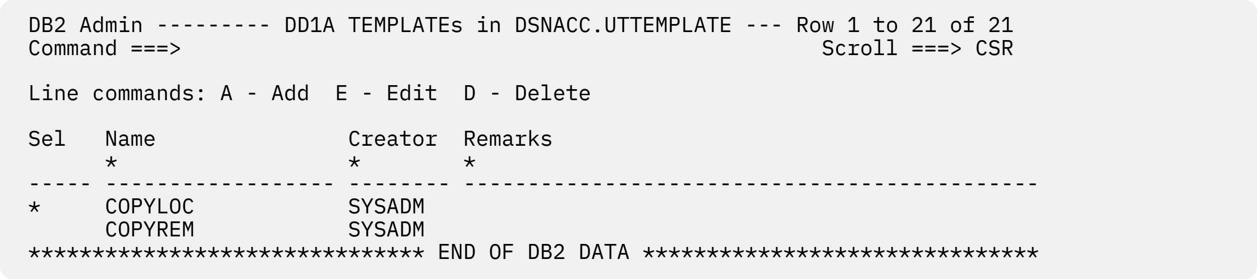


```
ADB25 min ----- DD1A Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

   L - Manage LISTDEFs                                    DB2 System: DD1A
   T - Manage TEMPLATEs                                   DB2 SQL ID: ADM001
  TU - Specify TEMPLATE usage

  CL - Create LISTDEF control table
  UL - Upgrade LISTDEF control table
  CT - Create TEMPLATE control table
  UT - Upgrade TEMPLATE control table

 LISTDEF control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTLIST            >

 TEMPLATE control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTTEMPLATE        >
```

*Figure 224. **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel*

2. Specify option T, and press Enter.

   The **TEMPLATEs (ADB25T)** panel is displayed, as shown in the following figure:



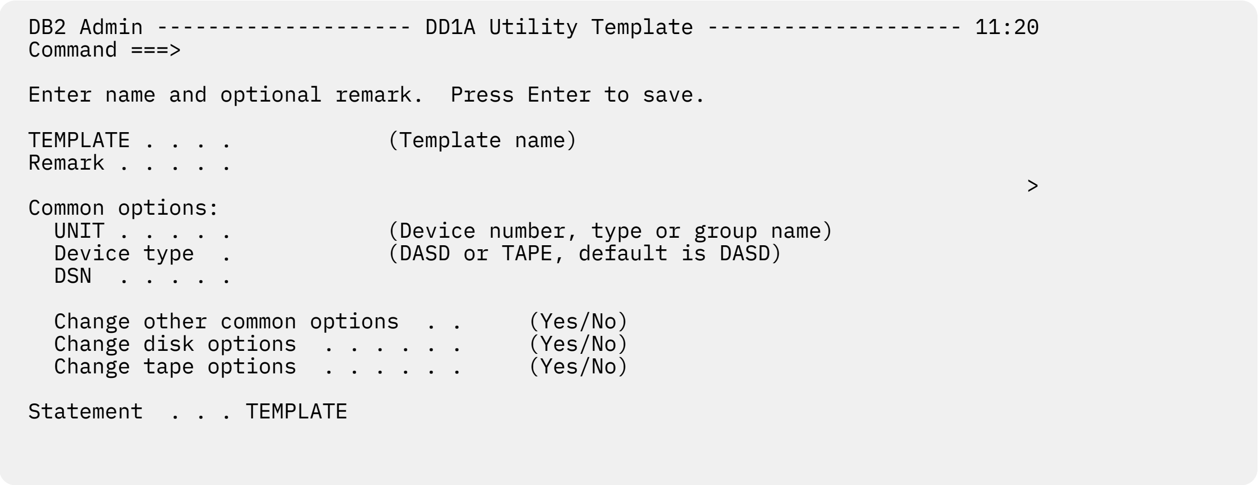
```
DB2 Admin --------- DD1A TEMPLATEs in DSNACC.UTTEMPLATE --- Row 1 to 21 of 21
Command ===>                                                  Scroll ===> CSR

Line commands: A - Add  E - Edit  D - Delete

Sel   Name               Creator  Remarks
      *                  *        *
----- ------------------ -------- ---------------------------------------------
*     COPYLOC            SYSADM
      COPYREM            SYSADM
******************************* END OF DB2 DATA *******************************
```

*Figure 225. **TEMPLATEs (ADB25T)** panel*

   This panel displays the existing templates within the control table. The table name is in the panel header. For example, in the previous figure, the table name is DSNACC.UTTEMPLATE.

3. Specify the E line command next to the template that you want to edit, and press Enter.

   The **Utility Template (ADB25TE)** panel is displayed, as shown in the following figure:



```
DB2 Admin -------------------- DD1A Utility Template -------------------- 11:20
Command ===>

Enter name and optional remark.  Press Enter to save.

TEMPLATE . . . .            (Template name)
Remark . . . . .
                                                                              >
Common options:
  UNIT . . . . .            (Device number, type or group name)
  Device type  .            (DASD or TAPE, default is DASD)
  DSN  . . . . .

  Change other common options  . .     (Yes/No)
  Change disk options  . . . . . .     (Yes/No)
  Change tape options  . . . . . .     (Yes/No)

Statement  . . . TEMPLATE
```

Figure 226. Utility Template panel (ADB25TE)

4. Modify any of the following fields:

TEMPLATE
A name for the template. The name must be unique within the control table that you are using.

DSN
A data set name pattern for the template. The data set name can be composed of variables whose value is determined and substituted during execution of the utility that is using the template or execution of the job that Db2 Admin Tool generated for alter, restore, redefine, migrate, or object comparison processing that is using the template.

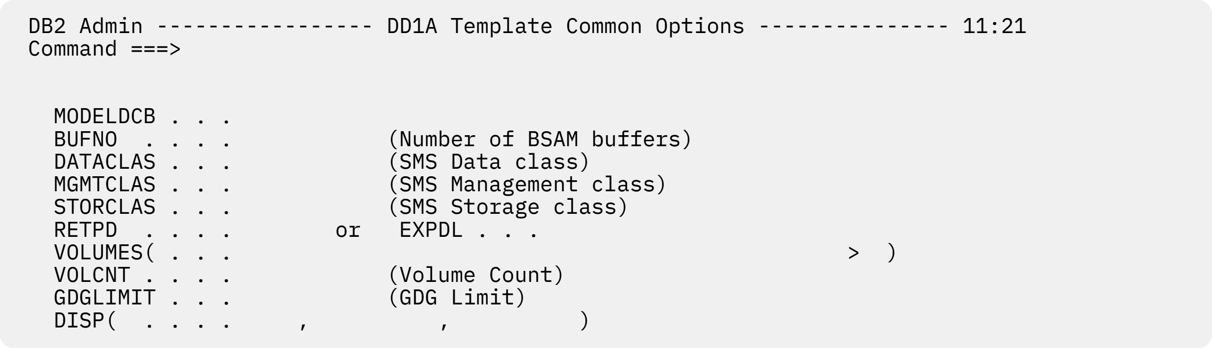
To construct a data set name pattern by using substitution variables, specify a question mark (?) as the first character of the **DSN** field. When you press Enter, the **Utility Template – Data Set Name (ADB25TD)** panel is displayed. For more information about data set names and substitution variables, see "Creating a template DSN by selecting substitution variables" on page 435

The variables displayed on the **Utility Template – Data Set Name (ADB25TD)** panel are the variables that are supported for normal Db2 utility template processing. Therefore, any variable displayed is valid for the data set name pattern for a utility data set template. However, not all of the variables are valid for the templates for non-utility work data sets, and additional variables might apply.

UNIT
The device number or group name for the data set.

Device type
The device type for the data set.

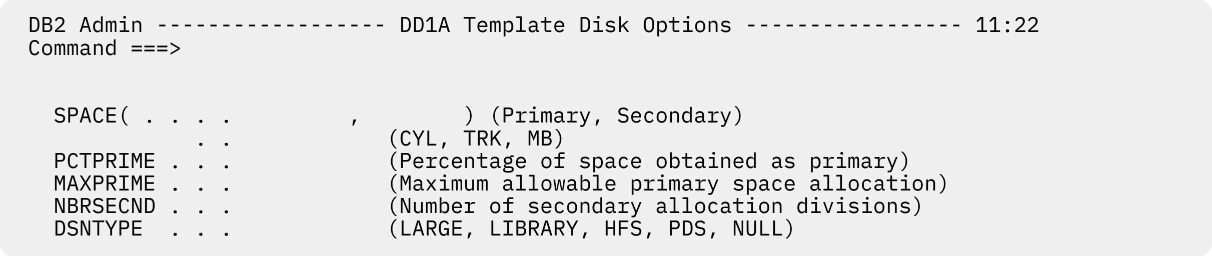
Change other common options
Additional attributes for the data set. When you specify Yes and press Enter, the **Template Common Options (ADB25TC)** panel is displayed, as shown in the following figure. See the online help for the description of the fields on this panel.



```
DB2 Admin ----------------- DD1A Template Common Options --------------- 11:21
Command ===>


  MODELDCB . . .
  BUFNO  . . . .            (Number of BSAM buffers)
  DATACLAS . . .            (SMS Data class)
  MGMTCLAS . . .            (SMS Management class)
  STORCLAS . . .            (SMS Storage class)
  RETPD  . . . .        or   EXPDL . . .
  VOLUMES( . . .                                                   >  )
  VOLCNT . . . .            (Volume Count)
  GDGLIMIT . . .            (GDG Limit)
  DISP(  . . . .     ,          ,          )
```

Figure 227. Template Common Options panel (ADB25TC)

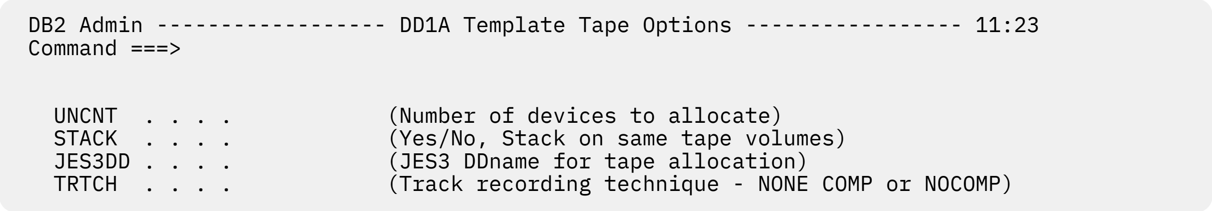
Change disk options
Additional options for the data set that are applicable only to data sets that are on disk. When you specify Yes and press Enter, the **Template Disk Options (ADB25TS)** panel is displayed, as shown in the following figure. See the online help for the description of the fields on the panel.



```
DB2 Admin ------------------ DD1A Template Disk Options ----------------- 11:22
Command ===>


  SPACE( . . . .         ,        ) (Primary, Secondary)
             . .            (CYL, TRK, MB)
  PCTPRIME . . .            (Percentage of space obtained as primary)
  MAXPRIME . . .            (Maximum allowable primary space allocation)
  NBRSECND . . .            (Number of secondary allocation divisions)
  DSNTYPE  . . .            (LARGE, LIBRARY, HFS, PDS, NULL)
```

*Figure 228. **Template Disk Options (ADB25TS)** panel*

Change tape options
Additional options for the data set that are applicable only for data sets on tape. When you specify Yes and press Enter, the **Template Tape Options (ADB25TT)** panel is displayed, as shown in the following figure. See the online help for the description of the fields on the panel.



```
DB2 Admin ------------------ DD1A Template Tape Options ----------------- 11:23
Command ===>


  UNCNT  . . . .            (Number of devices to allocate)
  STACK  . . . .            (Yes/No, Stack on same tape volumes)
  JES3DD . . . .            (JES3 DDname for tape allocation)
  TRTCH  . . . .            (Track recording technique - NONE COMP or NOCOMP)
```

*Figure 229. **Template Tape Options (ADB25TT)** panel*

5. Press Enter.

   Your changes are saved.

Deleting templates

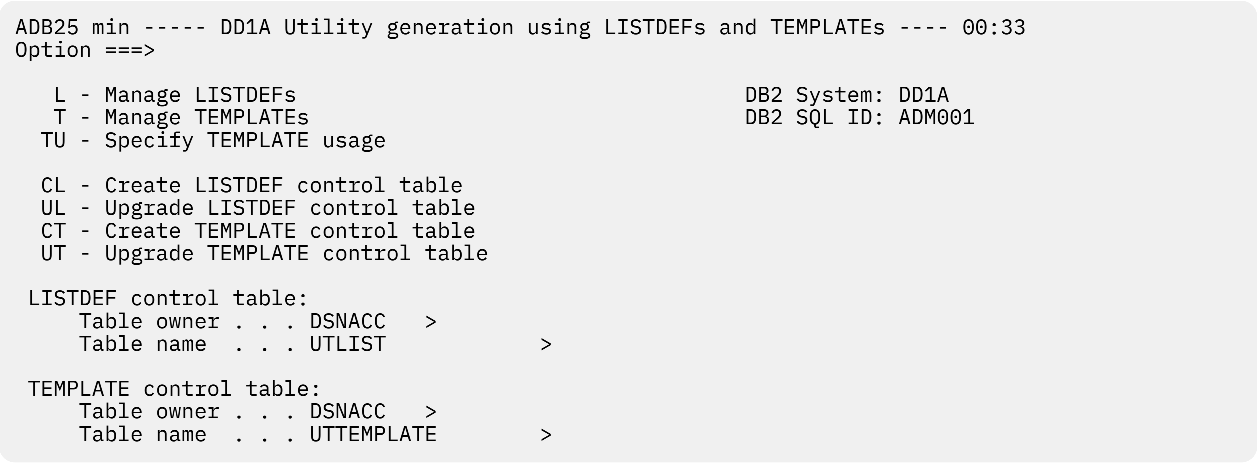
You can delete data set templates that you no longer want to use.

Procedure

To delete a template:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 5, and press Enter.

   The **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel is displayed.

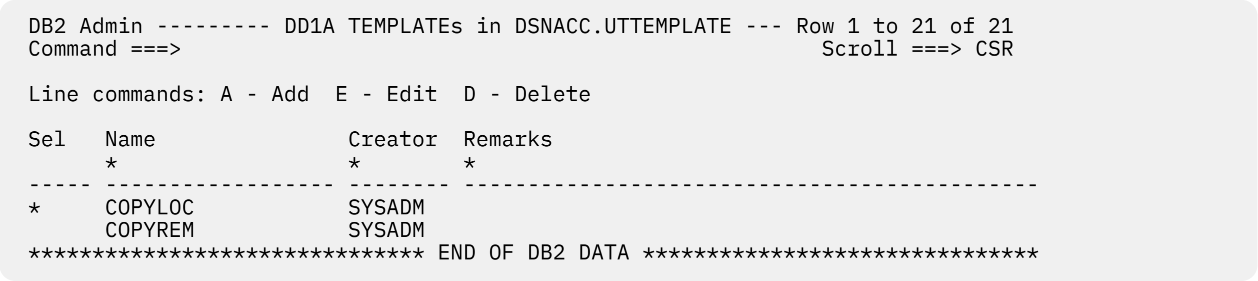


```
ADB25 min ----- DD1A Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

   L - Manage LISTDEFs                                   DB2 System: DD1A
   T - Manage TEMPLATEs                                  DB2 SQL ID: ADM001
  TU - Specify TEMPLATE usage

  CL - Create LISTDEF control table
  UL - Upgrade LISTDEF control table
  CT - Create TEMPLATE control table
  UT - Upgrade TEMPLATE control table

 LISTDEF control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTLIST            >

 TEMPLATE control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTTEMPLATE        >
```

*Figure 230. **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel*

2. Specify option T, and press Enter.

   The **TEMPLATEs (ADB25T)** panel is displayed, as shown in the following figure:



```
DB2 Admin --------- DD1A TEMPLATEs in DSNACC.UTTEMPLATE --- Row 1 to 21 of 21
Command ===>                                                 Scroll ===> CSR

Line commands: A - Add  E - Edit  D - Delete

Sel   Name              Creator  Remarks
      *                 *        *
----- ----------------- -------- --------------------------------------------
*     COPYLOC           SYSADM
      COPYREM           SYSADM
****************************** END OF DB2 DATA *******************************
```

*Figure 231. **TEMPLATEs (ADB25T)** panel*

   It shows the existing templates within the control table. The table name is in the panel header. For example, in the previous figure, the table name is DSNACC.UTTEMPLATE.

3. Specify option D in the Sel column that corresponds to the template that you want to delete, and press Enter.

   The template is immediately deleted.

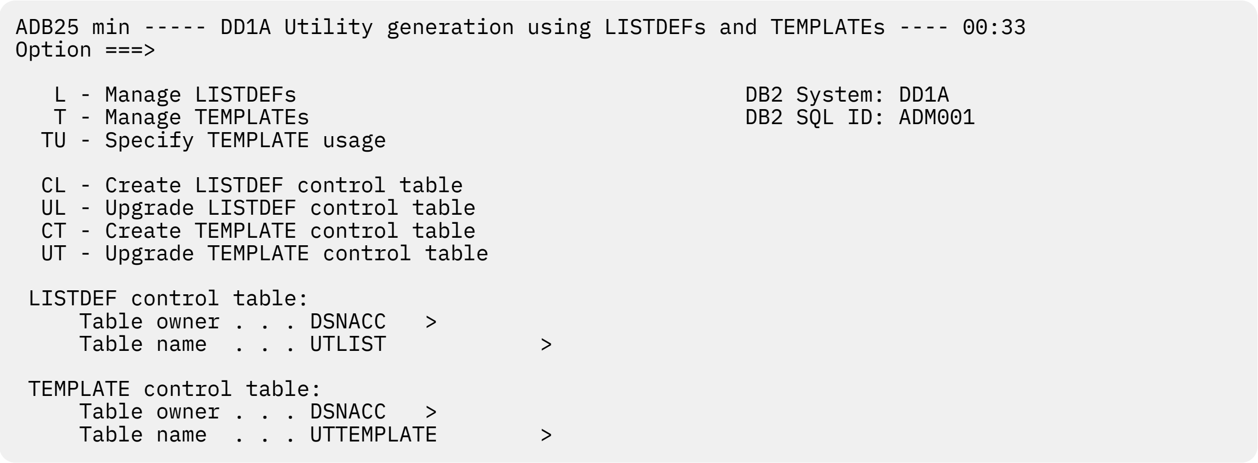
Upgrading the TEMPLATE control tables

Use this information to upgrade a TEMPLATE control table to the current Db2 version.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 5, and press Enter.

   The **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel is displayed.



```
ADB25 min ----- DD1A Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

   L - Manage LISTDEFs                                   DB2 System: DD1A
   T - Manage TEMPLATEs                                  DB2 SQL ID: ADM001
  TU - Specify TEMPLATE usage

  CL - Create LISTDEF control table
  UL - Upgrade LISTDEF control table
  CT - Create TEMPLATE control table
  UT - Upgrade TEMPLATE control table

 LISTDEF control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTLIST            >

 TEMPLATE control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTTEMPLATE        >
```

*Figure 232. **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel*

2. Specify option UT, and press Enter.

   The table name is validated to make sure it is a TEMPLATE control table. The validation is based on the following column names and data types:

   - NAME VARCHAR(8),
   - CREATEDBY VARCHAR(8),
   - MODIFIEDBY VARCHAR(8),
   - DSN VARCHAR(254),
   - DISPSTATUS VARCHAR(3),
   - DISPNTERM VARCHAR(7),
   - DISPATERM VARCHAR(7),
   - DEVICETYPE VARCHAR(8),
   - MODELDCB VARCHAR(53),
   - BUFNO SMALLINT,
   - DATACLAS VARCHAR(8),
   - MGMTCLAS VARCHAR(8),
   - STORCLAS VARCHAR(8),
   - DSVOLSER VARCHAR(1784),
   - GDGLIMIT INTEGER,
   - EXPDL VARCHAR(10),
   - RETPD INTEGER,
   - UNITTYPE CHAR(1),
   - PQTY INTEGER,
   - SQTY INTEGER,
   - SPACEUNIT VARCHAR(3),
   - PCTPRIME INTEGER,
   - MAXPRIME INTEGER,
   - NBRSECND INTEGER,
   - UNCNT SMALLINT,
   - STACK CHAR(1),
   - JES3DD VARCHAR(8),
   - TRTCH VARCHAR(6),

- REMARKS VARCHAR(254),
- VOLCNT SMALLINT.

If the TEMPLATE control table name is not at the current version, the control table is upgraded.

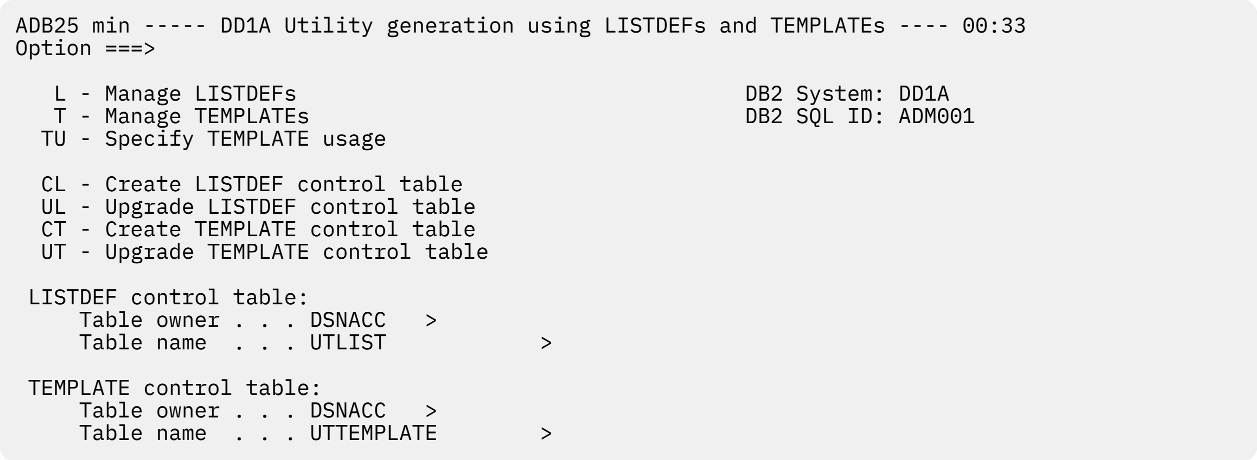
Creating a template DSN by selecting substitution variables

Use this information to learn how to create a template data set name by selecting the substitution variables to use.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 5, and press Enter.

   The **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel is displayed.

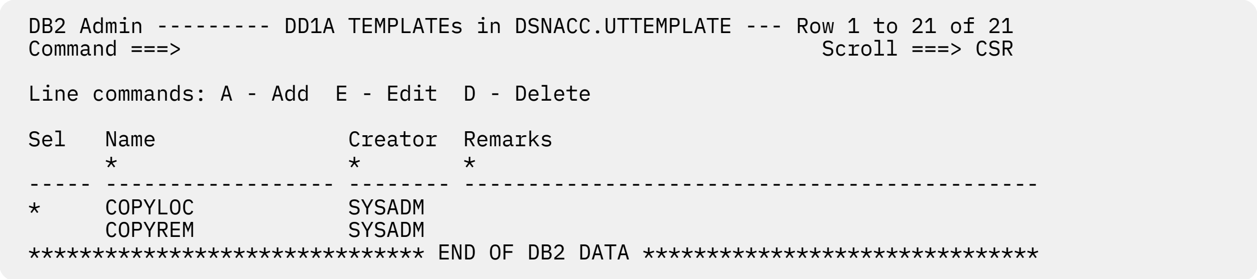


```
ADB25 min ----- DD1A Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

   L - Manage LISTDEFs                                  DB2 System: DD1A
   T - Manage TEMPLATEs                                 DB2 SQL ID: ADM001
  TU - Specify TEMPLATE usage

  CL - Create LISTDEF control table
  UL - Upgrade LISTDEF control table
  CT - Create TEMPLATE control table
  UT - Upgrade TEMPLATE control table

 LISTDEF control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTLIST            >

 TEMPLATE control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTTEMPLATE        >
```

*Figure 233. **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel*

2. Specify option T, and press Enter.

   The **TEMPLATEs (ADB25T)** panel is displayed, as shown in the following figure:



```
DB2 Admin --------- DD1A TEMPLATEs in DSNACC.UTTEMPLATE --- Row 1 to 21 of 21
Command ===>                                                  Scroll ===> CSR

Line commands: A - Add  E - Edit  D - Delete

Sel   Name               Creator  Remarks
      *                  *        *
----- ------------------ -------- ---------------------------------------------
*     COPYLOC            SYSADM
      COPYREM            SYSADM
******************************* END OF DB2 DATA *******************************
```

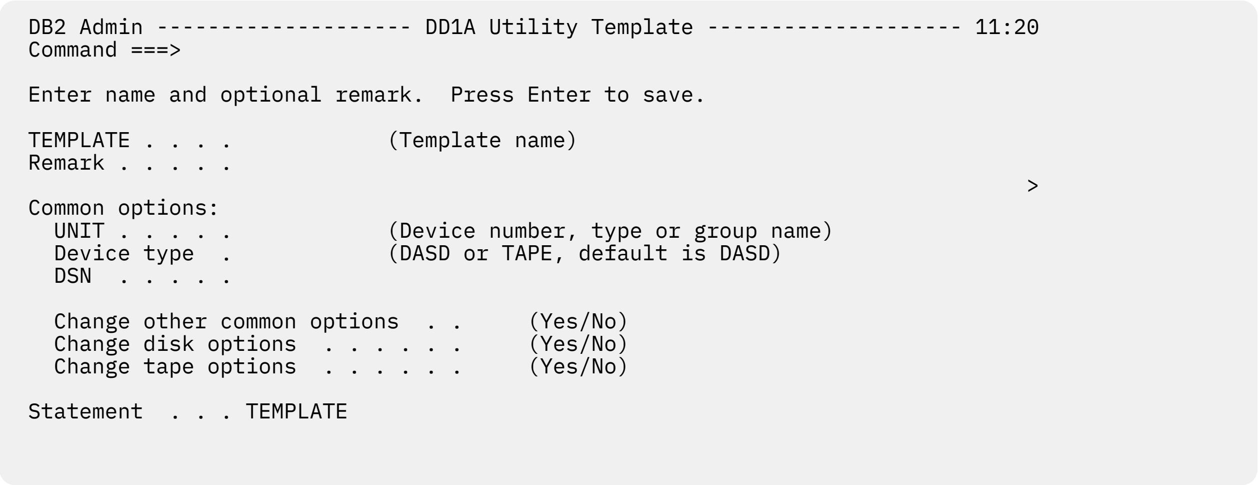
*Figure 234. **TEMPLATEs (ADB25T)** panel*

   It shows the existing TEMPLATEs within the control table. The table name is in the panel header. For example, in the previous figure, the table name is `DSNACC.UTTEMPLATE`.

3. Choose one of the following options:

   - If you are adding a new TEMPLATE, specify A in the Sel column of the first row, and press Enter.
   - If you want to edit a TEMPLATE, specify E in the Sel column that corresponds to the template that you want to edit, and press Enter.

   The **Utility Template (ADB25TE)** panel is displayed, as shown in the following figure:

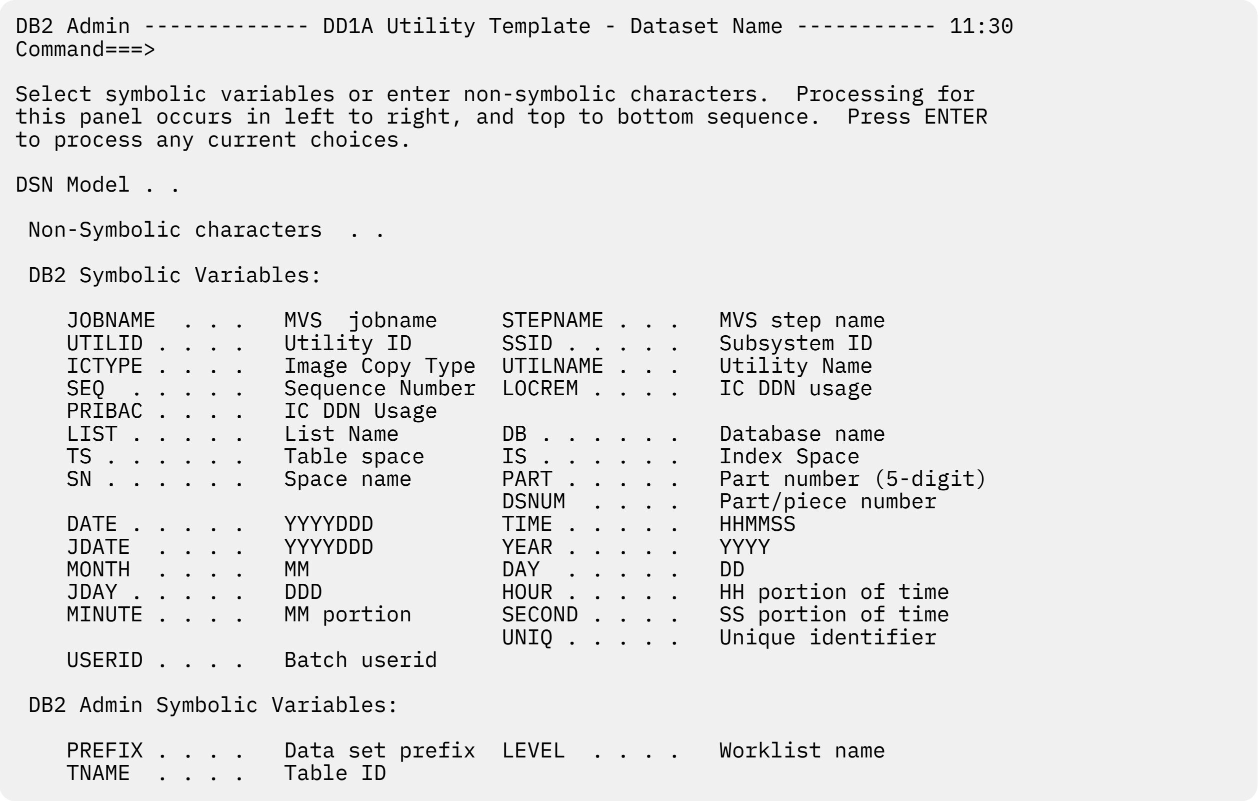


```
DB2 Admin -------------------- DD1A Utility Template -------------------- 11:20
Command ===>

Enter name and optional remark.  Press Enter to save.

TEMPLATE . . . .            (Template name)
Remark . . . . .
                                                                          >
Common options:
  UNIT . . . . .            (Device number, type or group name)
  Device type  .            (DASD or TAPE, default is DASD)
  DSN  . . . . .

  Change other common options  . .     (Yes/No)
  Change disk options  . . . . . .     (Yes/No)
  Change tape options  . . . . . .     (Yes/No)

Statement  . . . TEMPLATE
```

Figure 235. Utility Template panel (ADB25TE)

4. In the **DSN** field, specify ?.

   The **Utility Template – Data Set Name (ADB25TD)** panel is displayed, as shown in the following figure.



```
DB2 Admin ------------- DD1A Utility Template - Dataset Name ----------- 11:30
Command===>

Select symbolic variables or enter non-symbolic characters.  Processing for
this panel occurs in left to right, and top to bottom sequence.  Press ENTER
to process any current choices.

DSN Model . .

 Non-Symbolic characters  . .

 DB2 Symbolic Variables:

    JOBNAME  . . .   MVS  jobname     STEPNAME . . .   MVS step name
    UTILID . . . .   Utility ID       SSID . . . . .   Subsystem ID
    ICTYPE . . . .   Image Copy Type  UTILNAME . . .   Utility Name
    SEQ  . . . . .   Sequence Number  LOCREM . . . .   IC DDN usage
    PRIBAC . . . .   IC DDN Usage
    LIST . . . . .   List Name        DB . . . . . .   Database name
    TS . . . . . .   Table space      IS . . . . . .   Index Space
    SN . . . . . .   Space name       PART . . . . .   Part number (5-digit)
                                      DSNUM  . . . .   Part/piece number
    DATE . . . . .   YYYYDDD          TIME . . . . .   HHMMSS
    JDATE  . . . .   YYYYDDD          YEAR . . . . .   YYYY
    MONTH  . . . .   MM               DAY  . . . . .   DD
    JDAY . . . . .   DDD              HOUR . . . . .   HH portion of time
    MINUTE . . . .   MM portion       SECOND . . . .   SS portion of time
                                      UNIQ . . . . .   Unique identifier
    USERID . . . .   Batch userid

 DB2 Admin Symbolic Variables:

    PREFIX . . . .   Data set prefix  LEVEL  . . . .   Worklist name
    TNAME  . . . .   Table ID
```

Figure 236. Utility Template — Data Set Name panel (ADB25TD)

5. Specify substitution variables:

   - To specify non-symbolic characters, type them in the **Non-Symbolic characters** field. Press Enter to transfer and append the characters you entered to the DSN Model field near the top of the panel, which contains the template data set name pattern.
   - To select a symbolic variable, type any character (such as a slash) to the right of the leader dots. Press Enter to transfer your choices to the DSN model, which causes the variable name, followed by either one or two periods, to be appended to the DSN model statement. The first period marks the end of the variable name, not the end of the qualifier. If the preceding item is a variable, two periods

are required in succession to begin a new name segment (qualifier). The first period marks the end of the variable and the second period marks the beginning of the next qualifier.

The variable names are appended to the data set name template in left-to-right and top-to-bottom order each time Enter is pressed. To append an earlier variable after a later variable, first select the later variable and press Enter; then append the earlier variable.

6. Verify that the data set name in the **DSN Model** field contains the appropriate number of periods. Also, for variables that will return numeric characters, ensure that an alphabetic character (A to Z) or national character (# @ $) precedes the variable if it begins a qualifier. Type directly in the field to make any changes.

   Restriction: Not all the symbolic variables that are listed are valid variables for the data set name pattern for the templates for Db2 Admin Tool work data sets for alter, restore, redefine, migrate and object comparison processing, and additional variables might apply. To specify any additional variables that are not listed, use the **Non-Symbolic characters** field or type them directly into the **DSN Model** field.

Example

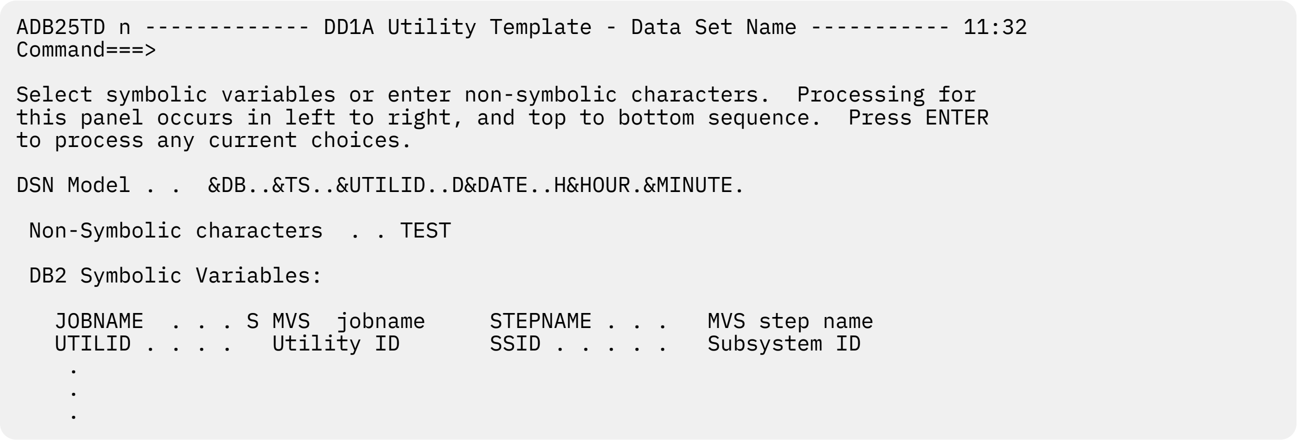
Example: `&JOBNAME..&STEPNAME.` displays two variables in succession. If the preceding item is a non-symbolic character and not a variable, only one period is used.

Example: In the example, `&JOBNAME.DSNCOPY`, no period follows DSNCOPY because it is the last qualifier and it is not a variable.

Example: In the example, `&USERID..D&DAY..M&MONTH..&DB(3,4).`, an alphabetic character precedes the variables DAY and MONTH because they return numeric characters. The use of substring notation on variables enables limiting the number of characters that are returned. Here, only four characters of the database name, starting at the third character, are returned.

Recommendation: Although it is permissible to enter variables in the DSN model by simply typing in the variables, use the panel fields to avoid spelling errors.

The example in the following figure uses the previous panel to show a partially completed DSN model statement; the non-symbolic TEST is about to be appended, followed by the *jobname* substitution variable.



```
ADB25TD n ------------- DD1A Utility Template - Data Set Name ----------- 11:32
Command===>

Select symbolic variables or enter non-symbolic characters.  Processing for
this panel occurs in left to right, and top to bottom sequence.  Press ENTER
to process any current choices.

DSN Model . .  &DB..&TS..&UTILID..D&DATE..H&HOUR.&MINUTE.

 Non-Symbolic characters  . . TEST

 DB2 Symbolic Variables:

   JOBNAME  . . . S MVS  jobname     STEPNAME . . .   MVS step name
   UTILID . . . .   Utility ID       SSID . . . . .   Subsystem ID
    .
    .
    .
```

Figure 237. Utility Template — Data Set Name example (ADB25TD) partial panel

Substitution variables in utility templates for PUNCHDDN

Typically, the template data set names for a utility are constructed by Db2 when the utility is processed, based on the template's data set name mask or pattern and substitution variables. However, when you use the Db2 Admin Tool functions for alter (ALT), migrate, rename database, and object comparison, the data set name that is associated with PUNCHDDN for a utility is resolved fully at JCL build time.

The data set name must be fully resolved and have valid qualifiers when the JCL is built because the data set for PUNCHDDN also becomes the input to the LOAD utility as the //SYSIN DD card. However, when the JCL is built for the data set name for PUNCHDDN, the value of some variables is unknown, and placeholder values are used instead. For example, if &JO or &JOBNAME is used as a substitution variable, JOBNAME is used as the value in the data set name.

The following table shows the replacement values for the symbolic variables that cannot be resolved at JCL build time for PUNCHDDN for (ALT), migrate, rename database, and object comparison:

Table 24. Replacement values for symbolic variables for templates for PUNCHDDN. Replacement values for symbolic variables for templates for PUNCHDDN

Symbolic variable	**Replacement value**
JOBNAME or JO	JOBNAME
UTILID	UTILID
STEPNAME	STEPNAME
SSID	The SSID
ICTYPE	ICTYPE
SEQ	SEQ
PRIBAC	PRIBAC
UTILNAME	UTILNAME
LOCREM	LOCREM
LIST	LIST
TS	The table space
SN	The table space
DB	The database name
IS	IS
PART	ALL
DATE	Build date in form YYYYDDD, for example, 2014190
JDATE	Julian date. Build date in form YYYYDDD, for example, 2014190
MONTH	The month, for example, 07
JDAY	The Julian day, for example, 190
MINUTE	The minutes, for example, 54
TIME	The time HHMMSS, for example, 135433
YEAR	The year, for example, 2014
DAY	The day, for example, 09
HOUR	The hour, for example, 13
SECOND	The seconds, for example, 33
USERID	The userid

Associating templates with data sets

You can associate templates with Db2 utility data sets and with Db2 Admin Tool work data sets.

About this task

Many Db2 utilities support the use of templates for data sets that are used by the utilities. Those data sets are identified in the utility statement by a DD name keyword. For example, REORG TABLESPACE has a DISCARDDN keyword to identify the data set to use for discarded records For these utility keywords, you can specify either a DD name or a template name to identify the data set. In Db2 Admin Tool, you can associate a template with these DD name keywords.

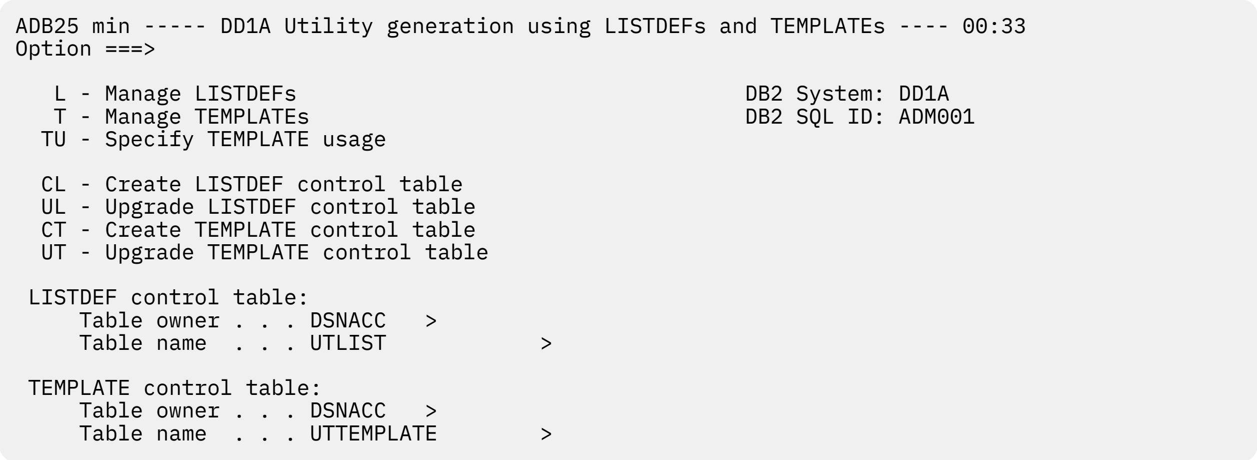
The Db2 Admin Tool work data sets that support the use of templates do so by using a template keyword. For example, the keyword ALDDL is used for the work data set that the ALT function uses for the DDL that is extracted from the catalog.

Procedure

To associate a template with a utility DD name keyword or a Db2 Admin Tooltemplate keyword:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 5, and press Enter.

   The **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel is displayed:

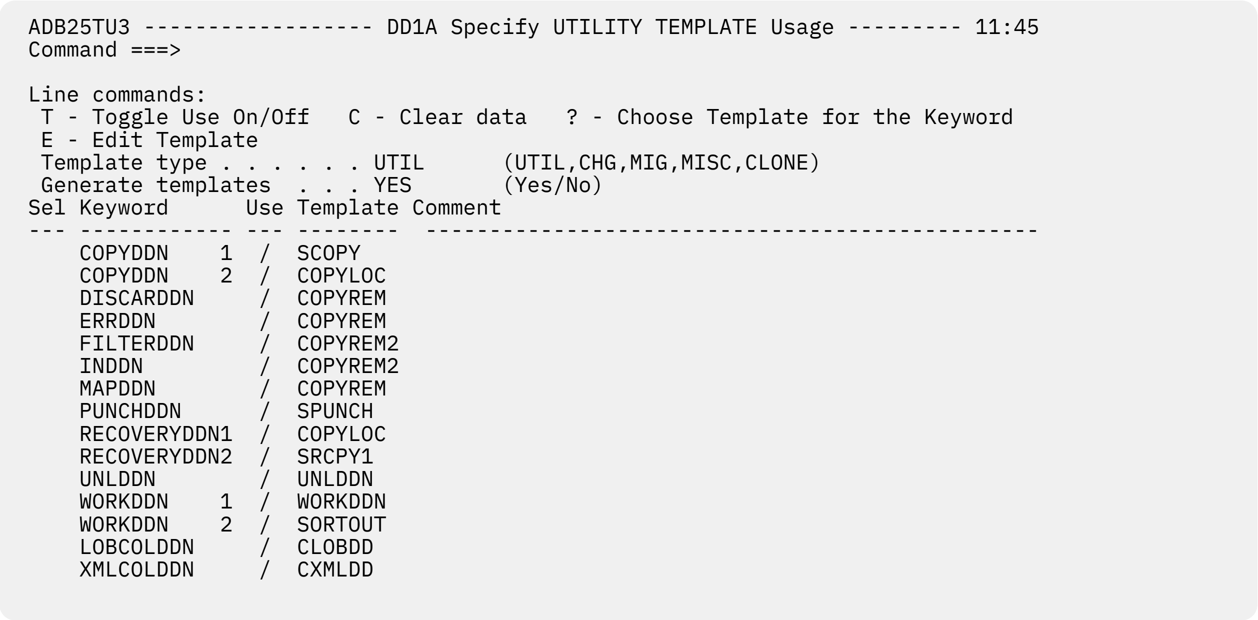


```
ADB25 min ----- DD1A Utility generation using LISTDEFs and TEMPLATEs ---- 00:33
Option ===>

   L - Manage LISTDEFs                                  DB2 System: DD1A
   T - Manage TEMPLATEs                                 DB2 SQL ID: ADM001
  TU - Specify TEMPLATE usage

  CL - Create LISTDEF control table
  UL - Upgrade LISTDEF control table
  CT - Create TEMPLATE control table
  UT - Upgrade TEMPLATE control table

 LISTDEF control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTLIST            >

 TEMPLATE control table:
     Table owner . . . DSNACC   >
     Table name  . . . UTTEMPLATE        >
```

   *Figure 238. **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel*

2. Specify option TU, and press Enter.

   The **Specify TEMPLATE Usage (ADB25TU3)** panel is displayed, as shown in the following figure. This panel indicates whether a template is actively associated with a keyword and the template name.



```
ADB25TU3 ------------------ DD1A Specify UTILITY TEMPLATE Usage --------- 11:45
Command ===>

Line commands:
 T - Toggle Use On/Off   C - Clear data   ? - Choose Template for the Keyword
 E - Edit Template
 Template type . . . . . . UTIL      (UTIL,CHG,MIG,MISC,CLONE)
 Generate templates  . . . YES       (Yes/No)
Sel Keyword      Use Template Comment
--- ------------ --- --------  ------------------------------------------------
    COPYDDN    1  /  SCOPY
    COPYDDN    2  /  COPYLOC
    DISCARDDN     /  COPYREM
    ERRDDN        /  COPYREM
    FILTERDDN     /  COPYREM2
    INDDN         /  COPYREM2
    MAPDDN        /  COPYREM
    PUNCHDDN      /  SPUNCH
    RECOVERYDDN1  /  COPYLOC
    RECOVERYDDN2  /  SRCPY1
    UNLDDN        /  UNLDDN
    WORKDDN    1  /  WORKDDN
    WORKDDN    2  /  SORTOUT
    LOBCOLDDN     /  CLOBDD
    XMLCOLDDN     /  CXMLDD
```

Figure 239. ***Specify TEMPLATE Usage (ADB25TU3)*** *panel*

If a keyword is listed twice (*keyword* 1 and *keyword* 2), that keyword allows two parameters. For example, in the previous screen, COPYDDN 1 corresponds to the first parameter and COPYDDN 2 corresponds to the second parameter.

For the UTIL keywords, a slash in the USE column indicates that a TEMPLATE statement is to be generated for any utility supporting templates for that keyword.

Note: For the CLONE template type, panel ADB25TU4 is displayed.

3. To change the template type that is displayed, overwrite the value in the **Template Type** field, and press Enter.

   The following values are valid:

   UTIL
   Utility data set keywords that are used by Db2 utilities.

   CHG
   Non-utility data set keywords that are used by the Db2 Admin Tool Alter (ALT) function, Db2 Object Comparison Tool, or Change Management.

   MIG
   Migrate data set keywords that are used by the Db2 Admin Tool Migrate function.

   MISC
   SYSPRINT data set keywords that are used by Db2 Admin Tool to generate work statement lists (WSLs) online.

   CLONE
   Utility templates that are used for cloned table spaces.

   For a description of the keywords for each of these types, see “Keywords that can be associated with templates” on page 441

4. To associate a template with a keyword, specify ? in the **SEL** column next to the keyword, and press Enter.

   The **TEMPLATEs (ADB25T)** panel displays a list of defined templates.

5. To select a template, specify a plus sign (+) in the **SEL** column next to the template name, and press Enter.

   The **Specify TEMPLATE Usage (ADB25TU3)** panel includes a slash (/) in the **Use** column next to the keyword for which you selected a template. The template name and any comments are also displayed.

Keywords that can be associated with templates

You can associate templates with data set keywords for utilities and various functions in Db2 Admin Tool.

Db2 Admin Tool categorizes templates into the following types: UTIL, CLONE, CHG, MIG, and MISC keywords (For a description of each of these templates types, see "Associating templates with data sets" on page 439.) The following sections describe the keywords with which you can associate templates for each of these template types.

UTIL and CLONE

For the UTIL and CLONE template types, you can associate templates with the following utility keywords:

FCCOPYDDN
The output data set for a FlashCopy image copy

COPYDDN 1
The output data set for the primary image copy.

COPYDDN 2
The output data set for the backup image copy

DISCARDDN
The discard data set

ERRDDN
A work data set that is used for error processing

FILTERDDN
A filter data set that is used by the COPY utility

INDDN
The input data set

MAPDDN
A work data set that is used by the LOAD utility for error processing

PUNCHDDN
A data set to which utility control statements are written

RECOVERYDDN 1
The data set for the primary copy at the recovery site

RECOVERYDDN 2
The data set for the backup copy at the recovery site

UNLDDN
The unload data set

WORKDDN 1
A temporary work data set

WORKDDN2
A second temporary work data set

LOBCOLDDN
The unload data set for LOBs

XMLCOLDDN
The unload data set for XML

When you specify the maximum data set name length, consider the following situations where the data set name might be extended:

- If templates are specified for the LOBCOLDDN, XMLCOLDDN, DISCARDDN, ERRDDN, WORKDDN 1, WORKDDN 2, or MAPDDN keywords and used in a change process, an additional &TNAME qualifier is added to the template if it is not already included. This qualifier extends the length of the result data set name by 6 bytes. Ensure that these added 6 bytes do not cause your data set names to be greater than the 44-byte limit.

- If a template is specified for the DISCARDDN keyword and used in a change process and the LOAD utility is to run in parallel in the APPLY batch job, an additional P&PA qualifier is added to the template if it is not already included. This qualifier extends the length of the result data set name by 7 bytes.

Both Db2 Admin Tool and Object Comparison Tool support the use of the REORG and COPY utilities in the Alter, Object Compare and CM functions. If COPYDDN 1 and COPYDDN 2 templates are specified, you should specify a unique symbolic variable to prevent conflicts.

CHG

For the CHG template type, you can associate templates with the following keywords:

ALCNT
Load control cards for the original objects

ALCNC
Load control cards for the altered objects

ALULD
Unloaded data from the original objects

ALULDC
Converted unloaded data

ALALTR
DDL for the altered objects

ALCREA
DDL for the created objects

ALDROP
DDL for the dropped objects

ALRBND
Db2 commands for the rebinds of plans and packages

ALREFR
DDL for the refresh of materialized query tables (MQT)

ALCMD
Db2 commands

ALIFF
An internal version file. This data set must be in PDS format with a DSN maximum length of 42 bytes.

ALSHVR
ISPF information that is used in the apply process. This data set must be in PDS format.

ALCHNG
File produced by the compare program to indicate changes.

ALMOD
Alternate prefix for the ALIFF, ALCHNG and ALSHVR data sets. The ALMOD template is used if templates for ALIFF, ALCHNG or ALSHVR are not specified. The DSN maximum length must be 34 bytes.

ALMTC
File to contain multi-target change information. This data set must be a sequential data set.

ALPUTIL
File to contain post-utility change information. This data set must be a sequential data set.

When you specify the maximum data set name length, consider the following situations where the data set name might be extended:

- If templates are specified for the ALCNT, ALCNC, ALULD or ALULDC keywords, an additional &TNAME qualifier is added to the template if it is not already included. This qualifier extends the length of the result data set name by 6 bytes.

- If templates are specified for the ALULD or ALULDC keywords and the UNLOAD is run in parallel in the APPLY batch job, an additional P&PA qualifier is added to the template if it is not already included. This qualifier extends the length of the result data set name by 7 bytes.

MIG

For the MIG template type, you can associate templates with the following keywords:

MIMLSIN
Intermediate data set

MIMLSOT
Merged template statement and load statement

MIUCONV
Work statement list data set

MISDROP
Drop DDL for the target system

MISQL
DDL for the target system

MIUOTHR
Work statement list data set

MI2WDD
Work statement list elements

MICTLIU
HPU LOAD control statements

MICTLOU
HPU converted template statements and load statements

MICTLOV
Converted template statements and load statements

MIDATVP
HPU unload partitioned object data file. The DSN maximum length must be 35 bytes.

MIDTVNP
HPU unload non-partitioned object data file

MIGCMD
File to hold REBIND output

MICOL
File to hold GENERATED ALWAYS records for identity columns

MIGSHVR
Partitioned data set to hold ISPF tables used to generate the MIG jobs in batch.

MITEMPL
Templates used by ADBTEP2 auto utilities.

MISC

For the MISC template type, you can associate a template with the following keyword:

ADBWORK
SYSPRINT data set to contain the output of Slows that are run online. The DSN maximum length must be 35 bytes.

User-defined or product default templates

You can use templates that you specify and product default templates.

User-defined templates
Templates that you specify. User-defined templates are in the ADBTEMPL DD data definition. For information about using symbol variables to specify Db2 TEMPLATE statements, see “Symbol variables in the ADBTEMPL file: Db2 TEMPLATE support” on page 796.

Product default template
Templates that are assigned by Db2 Admin Tool when you do not specify a template.

If you use a product default template, you must manually add the --#TEMPLATE comment statement in the WSL. For example, if the MAPDDN template is defined, add the following comment statement:

```
--#TEMPLATE UTLMAP TYPE(TAPE)
TEMPLATE UTLMAP DSN 'SYSADM.XXX.T001'
UNIT TAPE
```

If the user-defined templates WORKDDN, MAPDDN, and ERRDDN are on removal media devices, you do not need to add the SPACE keyword.

Unloading data from LOBs by using the utility template

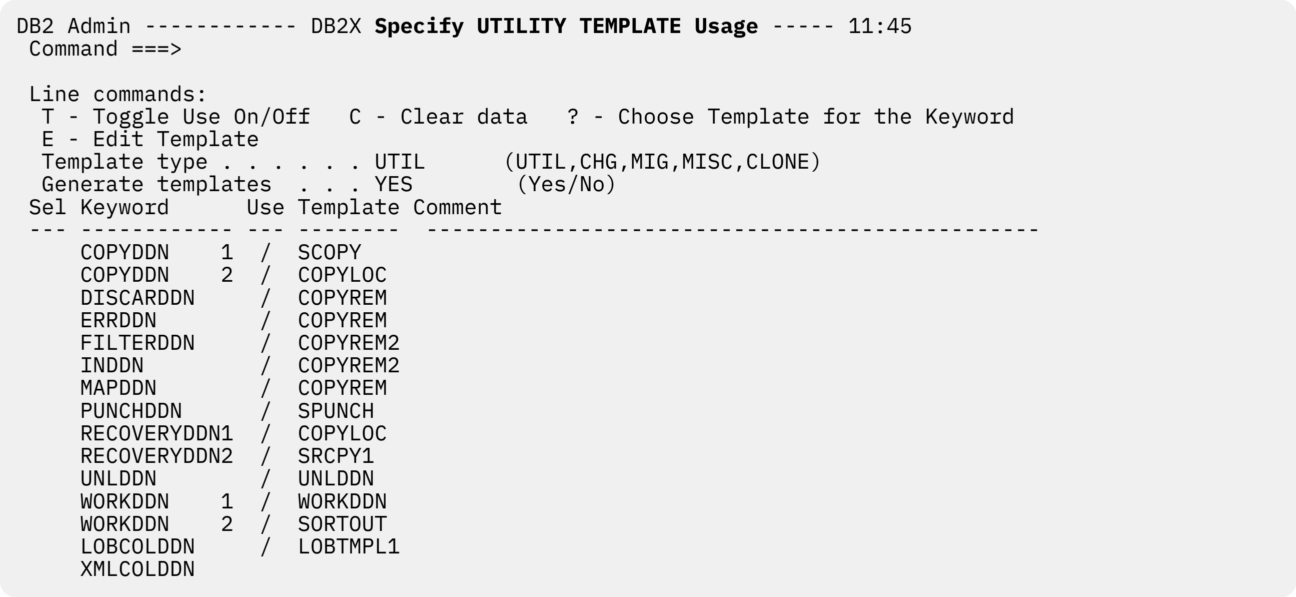
If you want to unload data from a LOB column, use a utility template.

When a table that contains multiple LOB columns needs to be unloaded, each LOB column requires a partitioned data set (PDS). You can use any utility-supported variables to define this template. The variables must be unique to ensure that data is not overwritten during unloads. If you do not specify a template, some functions, such as ALT and MIG, use the default template that Db2 Admin Tool assigns.

The utility template for LOBs uses the following process:

1. The function, such as ALT or MIG, generates the unload.
2. The utility template statements are added to the WSL.
3. The WSL runs, and ADBTEP2 converts the UNLOAD syntax before passing it to Db2.

To set up and use the utility template for LOBs, follow the steps in “Associating templates with data sets” on page 439. After you have associated the template name with the LOBCOLDDN keyword, the following panel is displayed.



```
DB2 Admin ------------ DB2X Specify UTILITY TEMPLATE Usage ----- 11:45
 Command ===>

 Line commands:
  T - Toggle Use On/Off   C - Clear data   ? - Choose Template for the Keyword
  E - Edit Template
  Template type . . . . . . UTIL      (UTIL,CHG,MIG,MISC,CLONE)
  Generate templates  . . . YES        (Yes/No)
 Sel Keyword      Use Template Comment
 --- ------------ --- --------  -----------------------------------------------
     COPYDDN    1  /  SCOPY
     COPYDDN    2  /  COPYLOC
     DISCARDDN     /  COPYREM
     ERRDDN        /  COPYREM
     FILTERDDN     /  COPYREM2
     INDDN         /  COPYREM2
     MAPDDN        /  COPYREM
     PUNCHDDN      /  SPUNCH
     RECOVERYDDN1  /  COPYLOC
     RECOVERYDDN2  /  SRCPY1
     UNLDDN        /  UNLDDN
     WORKDDN    1  /  WORKDDN
     WORKDDN    2  /  SORTOUT
     LOBCOLDDN     /  LOBTMPL1
     XMLCOLDDN
```

*Figure 240. **Specify UTILITY TEMPLATE Usage** panel (ADB25TU)*

Notes:

- The ADBL prefix is reserved for LOB template names that will be generated by the Run WSL function.
- The LOBCOLDDN data set name cannot exceed 35 bytes and must be a PDS.
- Do not specify a member name (for example, ADB.TEST.LOBCOL.OUT(MEMB2)).

Unloading data from an XML column by using the utility template

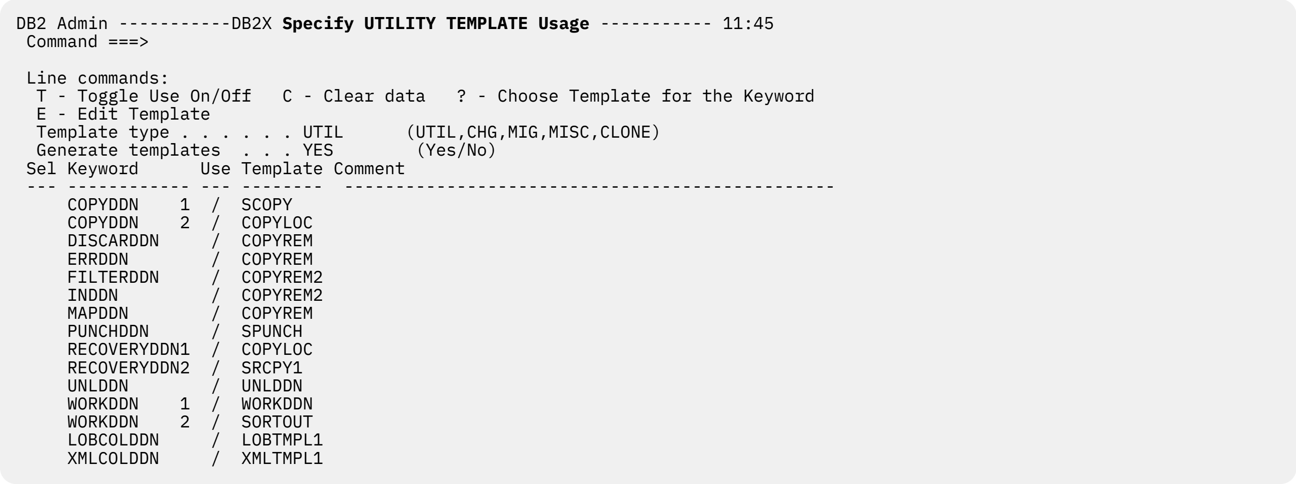
If you want to unload data from an XML column, use a utility template.

When a table that contains multiple XML columns needs to be unloaded, each XML column requires a partitioned data set (PDS). You can use any utility-supported variables to define this template. The resulting data set name that is built by using the variables must be unique to ensure that data is not overwritten. If you do not specify a template, the functions, such as ALT and MIG, use the default template that Db2 Administration Tool assigns.

The utility template for XML uses the following process:

1. The function (such as ALT and MIG) generates the unload.
2. The utility template statements are added to the WSL.
3. The WSL runs, and ADBTEP2 converts the UNLOAD syntax before passing it to Db2.

To set up and use the utility template for XML data, follow the steps in "Associating templates with data sets" on page 439. After you have associated the template name with the XMLCOLDDN keyword, the following panel is displayed.



```
DB2 Admin -----------DB2X Specify UTILITY TEMPLATE Usage ----------- 11:45
 Command ===>

 Line commands:
  T - Toggle Use On/Off   C - Clear data   ? - Choose Template for the Keyword
  E - Edit Template
  Template type . . . . . . UTIL      (UTIL,CHG,MIG,MISC,CLONE)
  Generate templates  . . . YES        (Yes/No)
 Sel Keyword      Use Template Comment
 --- ------------ --- --------  ------------------------------------------------
     COPYDDN    1  /  SCOPY
     COPYDDN    2  /  COPYLOC
     DISCARDDN     /  COPYREM
     ERRDDN        /  COPYREM
     FILTERDDN     /  COPYREM2
     INDDN         /  COPYREM2
     MAPDDN        /  COPYREM
     PUNCHDDN      /  SPUNCH
     RECOVERYDDN1  /  COPYLOC
     RECOVERYDDN2  /  SRCPY1
     UNLDDN        /  UNLDDN
     WORKDDN    1  /  WORKDDN
     WORKDDN    2  /  SORTOUT
     LOBCOLDDN     /  LOBTMPL1
     XMLCOLDDN     /  XMLTMPL1
```

*Figure 241. **Specify UTILITY TEMPLATE Usage** panel (ADB25TU)*

Notes:

- The ADBX prefix is reserved for XML template names that will be generated by the Run WSL function.
- The XMLCOLDDN data set name cannot exceed 35 bytes and must be a PDS.
- Do not specify a member name (for example, ADB.TEST.XMLCOL.OUT(MEMB2)).

Altering Db2 objects

Some object changes can be implemented by using an SQL ALTER statement. Other changes require that the object be dropped and recreated.

Procedure

To alter Db2 objects:

- Issue either the AL or the ALT line command, depending on the object and type of change.

Use the following table to determine which command to use and the panel on which to issue the command based on the object that you are changing.

Table 25. AL and ALT support for various objects

Object	Command supported (AL or ALT)	Panel
Column masks	AL	**Column Masks (ADB21PM)** panel
Databases	Both [1]	**Databases (ADB21D)** panel
Foreign keys	ALT	**Foreign Keys of Table (ADB21TFK)** panel
Functions	Both [1]	**Functions (ADB21F)** panel
Global variables	ALT	**Global Variables (ADBP1GV)** panel
Indexes	Both [1]	**Indexes (ADB21X)** panel
Row permissions	AL	**Row Permissions (ADB21PM)** panel
Sequences	Both [1]	**Sequence Objects (ADB21Q)** panel
Sequence aliases	Both [1]	**Sequence Objects (ADB21Q)** panel
Storage groups	AL	**Alter Storage Group (ADB21GA)** panel
Stored procedures	Both [1]	**Stored Procedures (ADB21O)** panel
Synonyms	ALT	**Synonyms (ADB21Y)** panel
Tables	Both [1]	**Tables, Views, and Aliases (ADB21T)** panel
Table aliases	Both [1]	**Aliases (ADB21A)** panel
Table spaces	Both [1]	**Table Spaces (ADB21S)** panel
Triggers	Both [1]	**Triggers (ADB21J)** panel
Trusted contexts	AL	**Trusted Contexts (ADB2AN)** panel
Views	ALT	**Tables, Views, and Aliases (ADB21T)** panel

Note:

1. If both AL and ALT are supported, use the following guidance to determine which command to use:
   - Use the AL line command for changes that can be made with SQL ALTER statements.
   - Use the ALT line command for changes that might require the object to be dropped and recreated, changes to multiple objects, and changes where you want to run Db2 utilities immediately after the change. The ALT line command can also make some, but not all, changes that are supported by ALTER statements.

- For general instructions on using ALT and AL, see the following topics:

  - “AL (Alter) function” on page 447
  - “ALT (Alter) function” on page 447
- For specific instructions on altering certain Db2 objects, see the following topics:
  - “Altering databases” on page 450
  - “Altering table spaces” on page 452
  - “Altering tables” on page 460
  - “Altering indexes” on page 490
  - “Altering triggers” on page 495
  - “Altering views” on page 497
  - “Altering foreign keys” on page 497
  - “Altering sequence aliases” on page 499

AL (Alter) function

When you use the AL line command to change an object, you invoke the AL function of Db2 Admin Tool. The AL function makes the requested object changes by executing an SQL ALTER statement.

The general steps for making changes with the AL line command are as follows:

1. Issue the AL line command against the object.
2. On the **ALTER *object*** panel, specify the changes that you want. This panel varies depending on the object.

   Subsequent panels might be displayed depending on your selections. Specify any additional information on those panels as needed.
3. Press Enter to run the ALTER statement.

For specific instructions for certain object types, see the following information:

- “Altering a database by using the AL line command” on page 450
- “Altering a table space by using the AL line command” on page 452
- “Altering a table by using the AL line command” on page 461
- “Altering an index by using the AL line command” on page 490
- “Altering triggers” on page 495

Related tasks

“Altering Db2 objects” on page 445
Some object changes can be implemented by using an SQL ALTER statement. Other changes require that the object be dropped and recreated.

ALT (Alter) function

When you use the ALT line command to change an object, you invoke the ALT function of Db2 Admin Tool. The ALT function first analyzes the change to determine what actions need to be taken to implement the change. You can choose whether you want this analysis to be done in batch or online.

The general steps for making changes with the ALT line command are as follows:

1. Issue the ALT line command against the object.
2. On the change panel for the object type, specify the change that you want to make, and issue the NEXT command.
3. On the **Alter Objects (ADB27CA)** panel, specify any other changes. If you want to change the method for analysis processing (batch or online), use the ALTOPT command.
4. Issue NEXT.

If you specified that analysis is to be done online, and the analysis process determines that SQL ALTER statements can accomplish the task, the **ADB27CTC** panel is displayed. On this panel, specify whether you want to run the SQL statements in the foreground (online) or generate a batch job:

- If you select ALTER statements, the SQL is run in the foreground. When the SQL completes successfully, the change is made, and you can skip the remaining steps.
- If you specify batch jobs, the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed.

Restriction: For accelerated tables where the change requires accelerator commands, running the SQL statements online is not allowed. In this situation, the **ADB27CTC** panel is not displayed and a batch job is always generated.

5. On the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel, specify the options for building the work statement list (WSL) or batch job to implement the change. You can also use the BP command to update the unit type and space parameters that are used for allocating the work and unload data sets.
6. Press Enter.

   An EDIT session is displayed with the generated job to analyze the change and build the apply jobs or WSL.

   Db2 Admin Tool generates the following jobs:

Job name	**Description**
ST1RE	Reverse engineering
ST1REALL	Reverse engineering when restore is enabled (optional)
ST2UL*nnn*	Unload data for table *nnn*
ST3AC	Convert
ST4AR	Alter related merge DDL (optional)
ST5DC	Drop and re-create
ST6RL*nnn*	Reload data
ST7CD	CHECK DATA (optional)
ST8IC*nnn*	Image copy
ST9RS	RUNSTATS (optional)
ST10RB	Rebind (optional)
ST11DL	Delete work data sets, except those data sets for restore and unload (optional)

   Db2 Admin Tool does not generate the conversion job step if it determines that data conversion is not needed.

   The numeric values in these job names are adjusted to occur sequentially if you omit optional steps or Db2 Admin Tool determines that the conversion step is not needed. For example, if related objects are not included, ST5DC becomes ST4DC.

7. Review the jobs and submit them in the sequence that they are listed to perform the changes.

   If one of the SQL statements in the input stream fails, you can use the Batch Restart program, ADBTEP2, to restart or resume the execution of an ALTER job at an intermediate point. In addition, you can combine the generated ALTER batch jobs into a single job.

   If you choose to have the statements that make the changes put in a WSL and specify that the WSL is to be generated online (instead of with a batch job), JCL to create the WSL is generated and run online. Messages are displayed to indicate the status as each step is run. When the online processing is complete, a work data set is displayed. This work data set contains the messages that would be seen in the job output if the WSL had been generated with a batch job.

Related tasks

“Altering Db2 objects” on page 445
Some object changes can be implemented by using an SQL ALTER statement. Other changes require that the object be dropped and recreated.

Related reference

“Work data sets used by the ALT (Alter Redefine) function” on page 449
When you redefine table spaces (or indexes) by using the ALT command, the Db2 Admin Tool redefine function creates and uses one or more data sets.

Related information

“The Batch Restart programs: ADBTEP2 and ADBTEPA” on page 572
The Batch Restart program, ADBTEP2, and the Authorization Switching program, ADBTEPA, are used with work statement lists and the Alter and Migrate Db2 data functions.

Work data sets used by the ALT (Alter Redefine) function

When you redefine table spaces (or indexes) by using the ALT command, the Db2 Admin Tool redefine function creates and uses one or more data sets.

The following table shows the data sets that are created and used by the redefine function.

Table 26. Work data sets that are used by the redefine function

Default data set name	Description	Template keyword
prefix.worklist.CREBIND	Rebind statements	RDBIND
prefix.worklist.DDL	DDL extracted from the catalog	RDSQL
prefix.worklist.DDDL	DDL to drop the original object	RDDROP
prefix.worklist.DDL.MOD	Modified DDL with the new table space or index statements or both	None[1]
prefix.worklist.CNTL.*database*.*tablespace*.PT	SYSPUNCH data set that is produced by the unload step	PUNCHDDN
prefix.worklist.UNLD.*database*.*tablespace*.PT	SYSREC data set that is produced by the unload step	UNLDDN
prefix.worklist.CNT.T*nnnn* where T*nnnn* is a string assigned to the data set, with *nnnn* beginning with 1	SYSPUNCH data set that is produced by the unload step when using High Performance Unload (HPU)	RDLOAD
prefix.worklist.ULD.T*nnnn* where T*nnnn* is a string assigned to the data set, with *nnnn* beginning with 1	SYSREC data set that is produced by the unload step when using High Performance Unload (HPU)	RDUNLD
prefix.worklist.UTRELD	LOAD utility statements when building a WSL	RDUTRELD
prefix.worklist.UTCHK	CHECK utility statements when building a WSL	RDUTCHK
prefix.worklist.UTRUNS	RUNSTATS utility statements when building a WSL	RDUTRUNS
prefix.worklist.UTCOPY	COPY utility statements when building a WSL	RDUTCOPY
prefix.worklist.UTUNLD	REORG and REBUILD utility statements when building a WSL	RDUTUNLD

Table 26. Work data sets that are used by the redefine function (continued)

Default data set name	Description	Template keyword
prefix.worklist.WDD	File names to include when building a WSL	RDWDD

Note:

1. The data set name is derived from the data set pattern that is specified for RDSQL. The data set name for RDSQL cannot be greater than 40 bytes.

Related concepts

"ALT (Alter) function" on page 447
When you use the ALT line command to change an object, you invoke the ALT function of Db2 Admin Tool. The ALT function first analyzes the change to determine what actions need to be taken to implement the change. You can choose whether you want this analysis to be done in batch or online.

Altering databases

You can change certain database attributes, including the name of the database. Depending on the change that you want to make, use either the AL or ALT line command.

About this task

Use the AL line command to make certain database changes that can made by the ALTER DATABASE statement. For example, you can use the AL line command to change the buffer pool, index buffer pool, or storage group.

Use the ALT line command for other changes. For example, to rename a database or to change the owner or CCSID for the database, you must use the ALT line command.

Note: In some cases, you can use either the ALT or AL line command. For example, although changing the buffer pool, index buffer pool, or storage group can be done with an ALTER statement, you can use either ALT or AL to make this change.

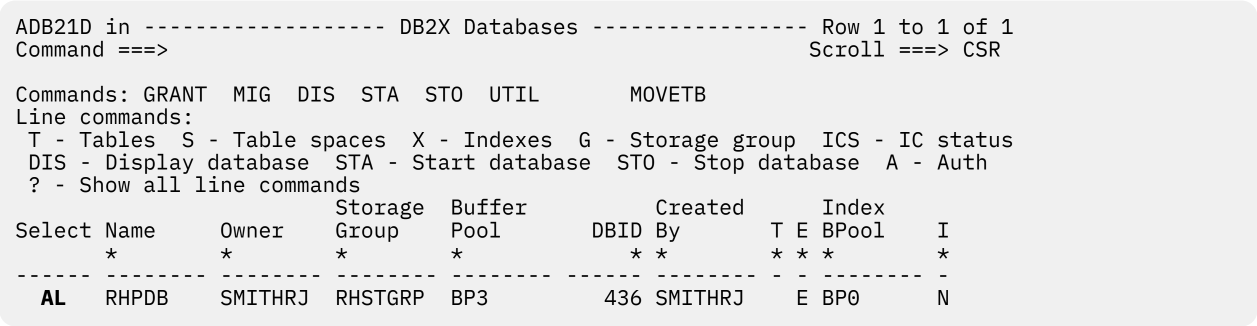
- "Altering a database by using the AL line command" on page 450
- "Altering a database by using the ALT line command" on page 451

Altering a database by using the AL line command

Procedure

To alter a database by using the AL line command:

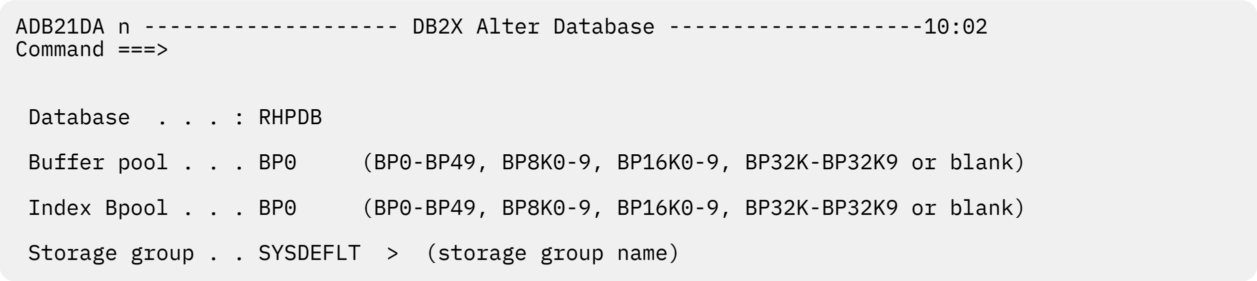
1. On the **DB2 Administration Menu (ADB2)** panel, select option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, select option D, and press Enter.
3. On the **Databases (ADB21D)** panel, specify the AL line command against the database that you want to alter, and press Enter:



```
ADB21D in ------------------- DB2X Databases ----------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       *        *        *        *             * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
  AL   RHPDB    SMITHRJ  RHSTGRP  BP3         436 SMITHRJ    E BP0      N
```

Figure 242. ***Databases (ADB21D)*** *panel*

4. On the **Alter Database (ADB21DA)** panel, specify new values for any fields that you want to change, and press Enter:



```
ADB21DA n -------------------- DB2X Alter Database --------------------10:02
Command ===>


 Database  . . . : RHPDB

 Buffer pool . . . BP0     (BP0-BP49, BP8K0-9, BP16K0-9, BP32K-BP32K9 or blank)

 Index Bpool . . . BP0     (BP0-BP49, BP8K0-9, BP16K0-9, BP32K-BP32K9 or blank)

 Storage group . . SYSDEFLT  >  (storage group name)
```

Figure 243. ***Alter Database (ADB21DA)*** *panel*

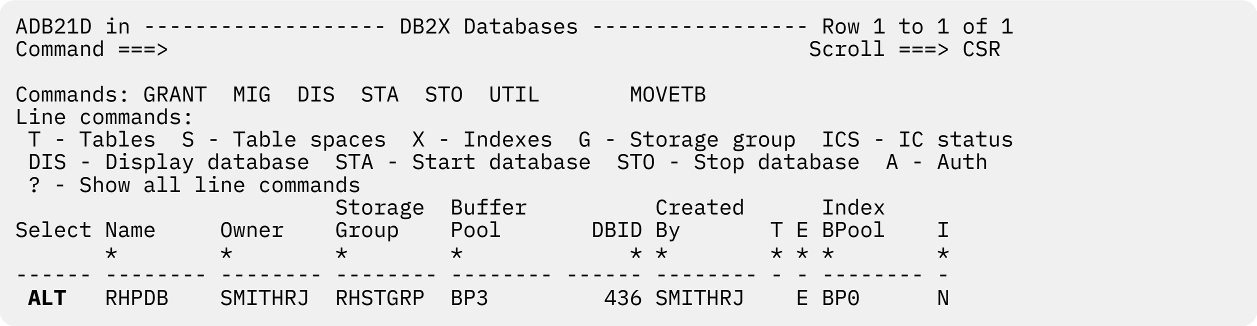
Db2 Admin Tool issues an ALTER DATABASE statement to make the changes.

Altering a database by using the ALT line command

Procedure

To alter a database by using the ALT line command:

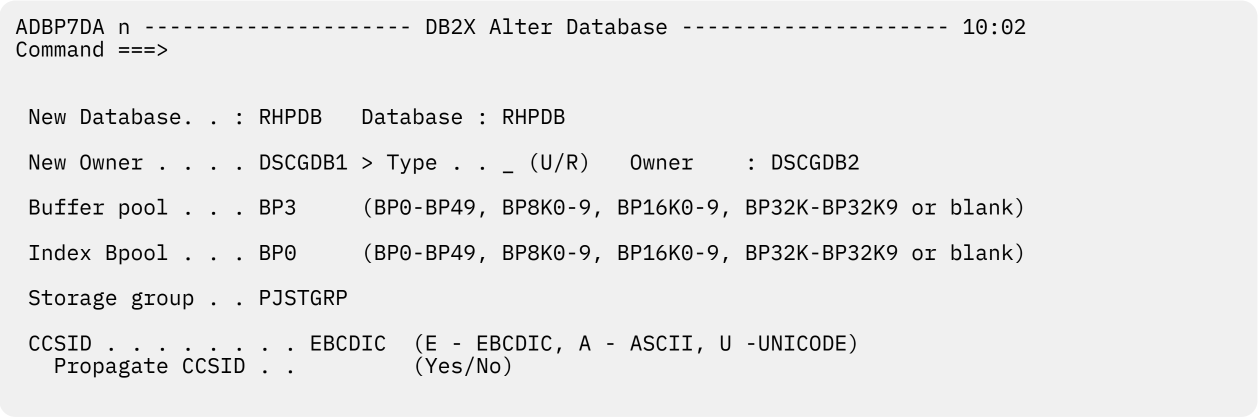
1. On the **DB2 Administration Menu (ADB2)** panel, select option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, select option D, and press Enter.
3. On the **Databases (ADB21D)** panel, specify ALT against the database that you want to change, and press Enter:



```
ADB21D in ------------------- DB2X Databases ----------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  UTIL       MOVETB
Line commands:
 T - Tables  S - Table spaces  X - Indexes  G - Storage group  ICS - IC status
 DIS - Display database  STA - Start database  STO - Stop database  A - Auth
 ? - Show all line commands
                         Storage  Buffer          Created      Index
Select Name     Owner    Group    Pool       DBID By       T E BPool    I
       *        *        *        *             * *        * * *        *
------ -------- -------- -------- -------- ------ -------- - - -------- -
 ALT   RHPDB    SMITHRJ  RHSTGRP  BP3         436 SMITHRJ    E BP0      N
```

Figure 244. ***Databases (ADB21D)*** *panel*

4. On the **Alter Database (ADBP7DA)** panel, specify any new values:
   - If you want a to rename the database, specify the new name in the **New database** field.
   - If you want to change the owner, specify that ID in the **New owner** field.
   - If you want to change the buffer pool, index buffer pool, storage group, or CCSID, specify the new values in the corresponding fields.



```
ADBP7DA n -------------------- DB2X Alter Database -------------------- 10:02
Command ===>


 New Database. . : RHPDB   Database : RHPDB

 New Owner . . . . DSCGDB1 > Type . . _ (U/R)   Owner    : DSCGDB2

 Buffer pool . . . BP3     (BP0-BP49, BP8K0-9, BP16K0-9, BP32K-BP32K9 or blank)

 Index Bpool . . . BP0     (BP0-BP49, BP8K0-9, BP16K0-9, BP32K-BP32K9 or blank)

 Storage group . . PJSTGRP

 CCSID . . . . . . . . EBCDIC  (E - EBCDIC, A - ASCII, U -UNICODE)
   Propagate CCSID . .         (Yes/No)
```

Figure 245. ***Alter Database (ADBP7DA)*** *panel*

5. Issue the NEXT command.
6. Generate an ALT job.

Altering table spaces

You can make changes to table spaces or partitions, such as changing the name or lock size or converting a table space to be a partitioned. Depending on the change that you want to make, use either the AL or ALT line command.

About this task

Use the AL line command to make table space changes that are supported by the ALTER TABLESPACE statement, such as changing the lock size or the maximum number of partitions.

Use the ALT line command to make table space changes that are not supported by the ALTER TABLESPACE statement, such as changing a non-partitioned table space to a partitioned table space. When you alter a table space with the ALT command, the table space is said to be *redefined*.

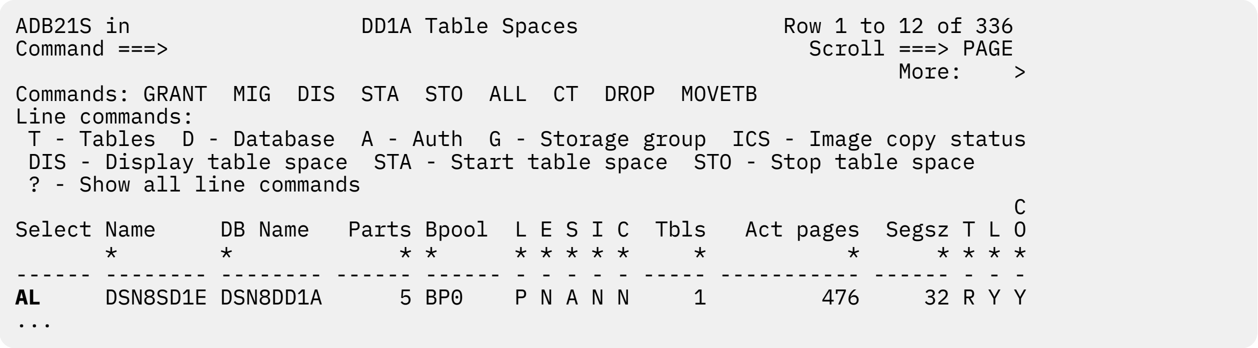
- “Altering a table space by using the AL line command” on page 452
- “Altering a table space by using the ALT command” on page 454

Altering a table space by using the AL line command

Procedure

To alter a table space by using the AL line command:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify any selection criteria at the bottom, specify option S, and press Enter.
3. On the **Table Spaces (ADB21S)** panel, specify the AL line command against the table space that you want to alter:



```
ADB21S in                  DD1A Table Spaces                Row 1 to 12 of 336
Command ===>                                                  Scroll ===> PAGE
                                                                     More:    >
Commands: GRANT  MIG  DIS  STA  STO  ALL  CT  DROP  MOVETB
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands
                                                                              C
Select Name     DB Name   Parts Bpool  L E S I C  Tbls   Act pages  Segsz T L O
       *        *             * *      * * * * *     *           *      * * * *
------ -------- -------- ------ ------ - - - - - ----- ----------- ------ - - -
AL     DSN8SD1E DSN8DD1A      5 BP0    P N A N N     1         476     32 R Y Y
...
```

4. On the **Alter Table Space (ADB21SA)** panel, change any table space or partition attributes, and press Enter.

   For partitioned table spaces, a detail line is displayed for each partition. You can alter any partition by updating an attribute, such as **FP**. To apply the same change to all partitions within the table space, provide a value on the **All Part** row.

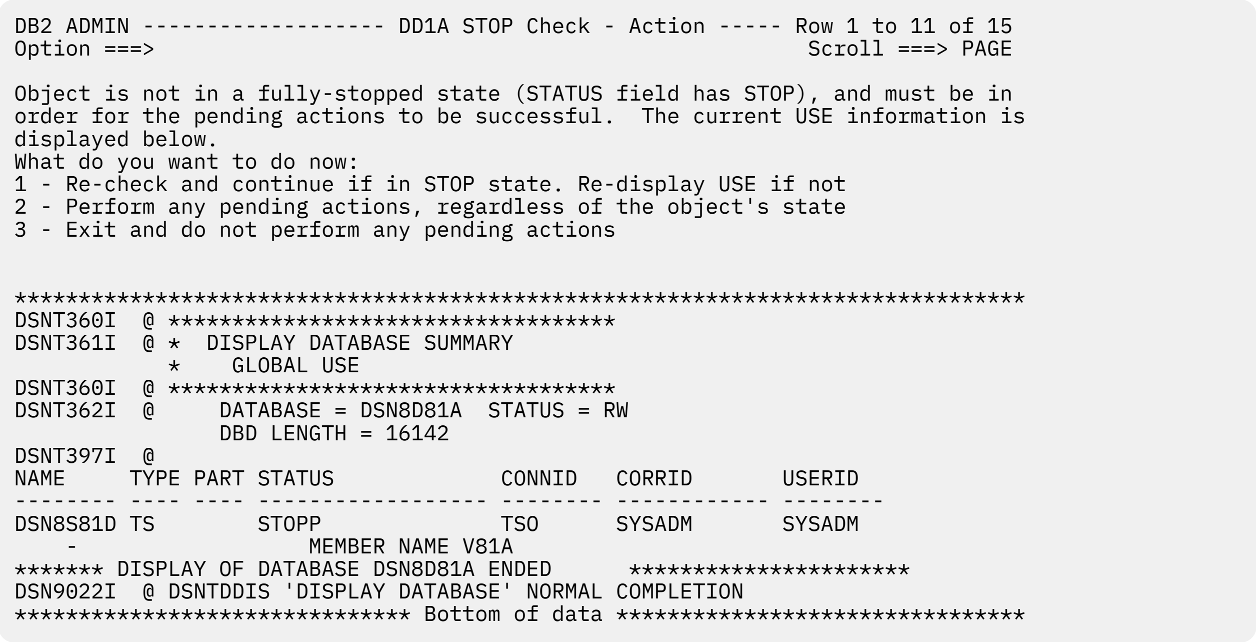
```
ADB21SA n -------------------- DD1A Alter Table Space -------- Row 1 to 5 of 5
Command ===>                                                    Scroll ===> PAGE


Line commands:
 D - Display Database   I - Interpret

ALTER TABLESPACE : DSN8DD1A.DSN8SD1E      (PBR - Partition by Range)
Buffer Pool  . . . . BP0        Close Rule  . . . NO  Max Rows  . . 255
Lock Size  . . . . . PAGE       Lock Part . . . . NO  Lock Max  . . SYSTEM
Max Partitions . . .            LOG . . . . . . . YES Insert Algo . 0
SEGSIZE  . . . . . . 32         MEMBER CLUSTER  . NO  PAGENUM . . .

                                         E T S
S   Part     Pqty    Sqty  FP PF PFU Cmp R M T VCAT     Stogroup GBPCach DSSIZE
       *        *       *   *  *   * *   * * * *        *        *       *
- ------ <------- <------ --- -- --- --- - - - -------- -------> ------- ------
All Part       -1      -1   0  5   0 YES N Y I DD1A     DSN8GD10 CHANGED
       1       -1      -1   0  5   0 YES N Y I DD1A     DSN8GD10 CHANGED
       2       -1      -1   0  5   0 YES N Y I DD1A     DSN8GD10 CHANGED
       3       -1      -1   0  5   0 NO  N Y I DD1A     DSN8GD10 CHANGED
       4       -1      -1   0  5   0 YES N Y I DD1A     DSN8GD10 CHANGED
       5       -1      -1   0  5   0 YES N Y I DD1A     DSN8GD10 CHANGED
```

   Note: Insert Algo is displayed only if you are running Db2 12 for z/OS.

   If the statement execution prompt is not enabled, an SQL ALTER TABLESPACE statement is executed with the parameters that you specified. If the **Statement Execution Prompt (ADB2PSTM)** panel is displayed, follow the instructions on that panel to complete and run the SQL statement to alter the table space.

   For changes to some parameters, such as PAGENUM, these changes are not immediately materialized; they are pending. You must reorganize the object for the change to take affect. For more information about which changes are pending, see Pending data definition changes (Db2 12 for z/OS).

   For other parameter changes, you must stop and restart the associated object. In these cases, Db2 Admin Tool issues a STOP DATABASE command for the table space, index, or partition. If the object is stopped, Db2 Admin Tool executes the ALTER TABLESPACE statement with the parameters that you specified and then a START DATABASE command to restart the stopped object. If the object is not in a fully-stopped state after Db2 Admin Tool issued the STOP DATABASE command, the **STOP Check - Action** panel prompts you to for the action to take:



```
DB2 ADMIN ------------------- DD1A STOP Check - Action ----- Row 1 to 11 of 15
Option ===>                                                  Scroll ===> PAGE

Object is not in a fully-stopped state (STATUS field has STOP), and must be in
order for the pending actions to be successful.  The current USE information is
displayed below.
What do you want to do now:
1 - Re-check and continue if in STOP state. Re-display USE if not
2 - Perform any pending actions, regardless of the object's state
3 - Exit and do not perform any pending actions


*******************************************************************************
DSNT360I  @ ***********************************
DSNT361I  @ *  DISPLAY DATABASE SUMMARY
            *    GLOBAL USE
DSNT360I  @ ***********************************
DSNT362I  @     DATABASE = DSN8D81A  STATUS = RW
                DBD LENGTH = 16142
DSNT397I  @
NAME     TYPE PART STATUS             CONNID   CORRID       USERID
-------- ---- ---- ------------------ -------- ------------ --------
DSN8S81D TS        STOPP              TSO      SYSADM       SYSADM
    -                  MEMBER NAME V81A
******* DISPLAY OF DATABASE DSN8D81A ENDED      **********************
DSN9022I  @ DSNTDDIS 'DISPLAY DATABASE' NORMAL COMPLETION
****************************** Bottom of data ********************************
```

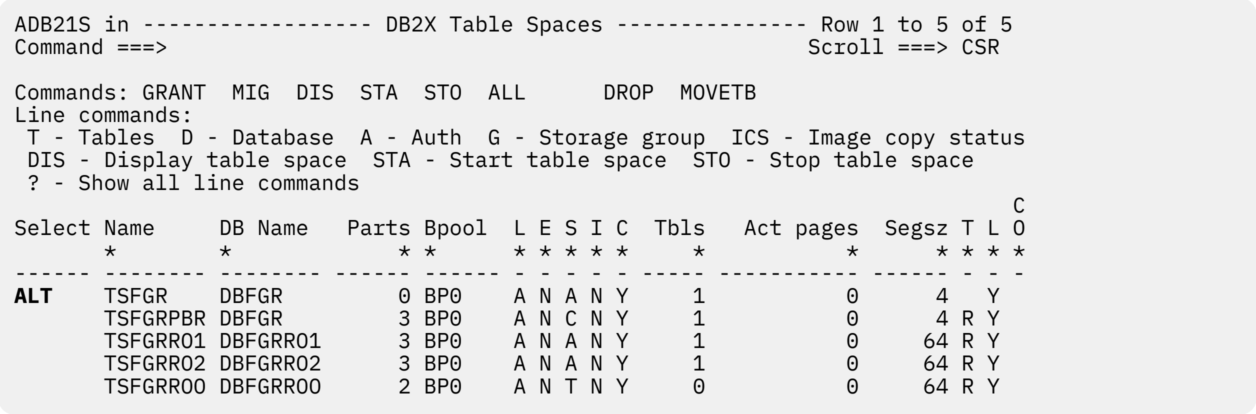
If an object is not stopped when the ALTER TABLESPACE statement runs, such as when others are holding locks on the object, a -626 SQL code is returned.

Altering a table space by using the ALT command

Procedure

To alter a table space by using the ALT command:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option S, and press Enter.
3. On the **Table Spaces (ADB21S)** panel, enter the ALT line command against the table space that you want to redefine, and press Enter:



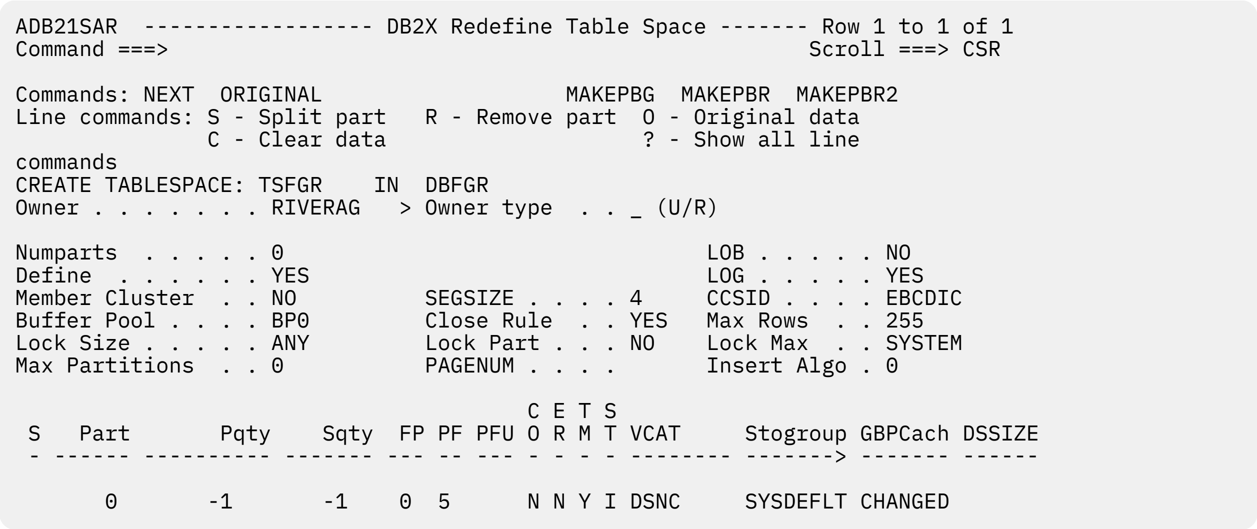
```
ADB21S in ------------------ DB2X Table Spaces --------------- Row 1 to 5 of 5
Command ===>                                                  Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL      DROP  MOVETB
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands
                                                                              C
Select Name     DB Name   Parts Bpool  L E S I C  Tbls   Act pages  Segsz T L O
       *        *             * *      * * * * *     *           *      * * * *
------ -------- -------- ------ ------ - - - - - ----- ----------- ------ - - -
ALT    TSFGR    DBFGR         0 BP0    A N A N Y     1           0      4   Y
       TSFGRPBR DBFGR         3 BP0    A N C N Y     1           0      4 R Y
       TSFGRRO1 DBFGRRO1      3 BP0    A N A N Y     1           0     64 R Y
       TSFGRRO2 DBFGRRO2      3 BP0    A N A N Y     1           0     64 R Y
       TSFGRROO DBFGRROO      2 BP0    A N T N Y     0           0     64 R Y
```

4. If the **Change Management Prompt (ADB2CMRO)** panel is displayed, specify whether you want to use Change Management, and press Enter.
5. On the **Redefine Table Space (ADB21SAR)** panel, change the table space parameters as needed and enter NEXT on the command line.

   Tips:

   - If you want to add or insert a partition, change the NUMPARTS value and issue the NEXT command. Then on the **Alter Partitioned Table (ADB21TAV)** panel, use the INS line command to insert any partitions and specify the limit key values.

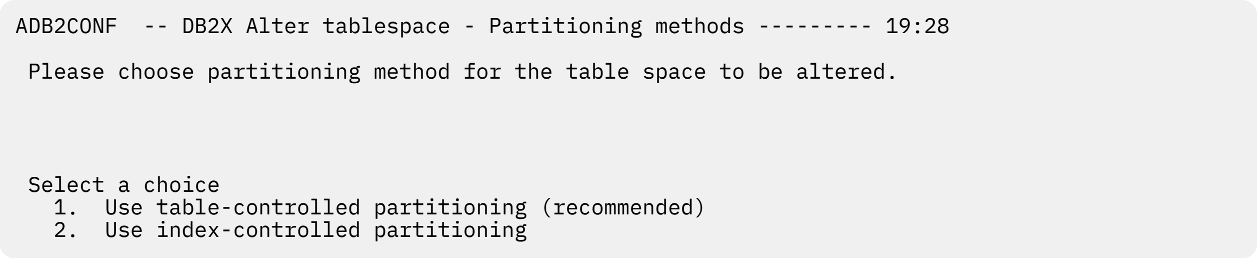
- If you want to change the table space type to partition-by-growth (PBG), use the MAKEPBG command.
- If you want to change the table space type to partition-by-range (PBR), use the MAKEPBR2 command or the MAKEPBR command. (MAKEPBR2 specifies relative page numbering; MAKEPBR specifies absolute page numbering.) On the subsequent **Alter Table (ADB21TAP)** panel, you can specify the partitioning key.



```
ADB21SAR  ------------------ DB2X Redefine Table Space ------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Commands: NEXT  ORIGINAL                   MAKEPBG  MAKEPBR  MAKEPBR2
Line commands: S - Split part   R - Remove part  O - Original data
               C - Clear data                    ? - Show all line
commands
CREATE TABLESPACE: TSFGR    IN  DBFGR
Owner . . . . . . . RIVERAG   > Owner type  . . _ (U/R)

Numparts  . . . . . 0                                 LOB . . . . . NO
Define  . . . . . . YES                               LOG . . . . . YES
Member Cluster  . . NO          SEGSIZE . . . . 4     CCSID . . . . EBCDIC
Buffer Pool . . . . BP0         Close Rule  . . YES   Max Rows  . . 255
Lock Size . . . . . ANY         Lock Part . . . NO    Lock Max  . . SYSTEM
Max Partitions  . . 0           PAGENUM . . . .       Insert Algo . 0

                                        C E T S
 S   Part       Pqty    Sqty  FP PF PFU O R M T VCAT     Stogroup GBPCach DSSIZE
 - ------ ---------- ------- --- -- --- - - - - -------- -------> ------- ------

       0       -1       -1    0  5      N N Y I DSNC     SYSDEFLT CHANGED
```

Note: Insert Algo is displayed only if you are running Db2 12 for z/OS.

If you are converting a segmented table space to a partitioned table space, the **Alter tablespace - Partitioning methods (ADB2CONF)** panel is displayed where you can select the partitioning method:



```
ADB2CONF  -- DB2X Alter tablespace - Partitioning methods --------- 19:28

 Please choose partitioning method for the table space to be altered.




 Select a choice
   1.  Use table-controlled partitioning (recommended)
   2.  Use index-controlled partitioning
```

Recommendation: Use table-controlled partitioning.

If you select option 1, the **Alter Table (ADB21TA)** panel is displayed. Specify the partitioning key for defining the table partitions.

If the **ALT - Index-controlled Partitioning (ADB21XAP)** panel is displayed, you can re-define an existing non-partitioning index to a partitioning index. If the **Create Partitioning Index (ADB21SAX)** panel is displayed, you can create a partitioning index.

6. Generate an ALT job.

Example

The following topics show specific examples of altering table spaces:

- “Reducing the MAXPARTITIONS value for a partition-by-growth table space” on page 456
- “Redefining partitions in a partitioned table space that uses index-controlled partitioning” on page 456
- “Redefining partitions in a partitioned table space that uses table-controlled partitioning” on page 459

Reducing the MAXPARTITIONS value for a partition-by-growth table space

You can use either the AL line command or the ALT line command to change the MAXPARTITIONS value of a partition-by-growth (PBG) table space. In both situations, an ALTER TABLESPACE statement is used to change the table space; the table space is not dropped and recreated.

AL method

Procedure

To reduce the MAXPARTITIONS value by using the AL line command:

1. On the **Table Spaces (ADB21S)** panel, issue the AL line command against the table space for which you want to reduce the MAXPARTITIONS value.
2. On the **Alter Table Space (ADB21SA)** panel, type a new value in the **Max Partitions** field, and press Enter.

   An ALTER TABLESPACE statement is executed and the MAXPARTITIONS value is reduced.

ALT method

Procedure

To reduce the MAXPARTITIONS value by using the ALT line command:

1. On **Table Spaces (ADB21S)** panel, issue the ALT line command against the table space for which you want to reduce the MAXPARTITIONS value.
2. On the **Redefine Table Space (ADB21SAR)** panel, type a new value in the **Max Partitions** field, and enter NEXT on the command line.
3. Generate an ALT job.

   In this case, Db2 Admin Tool determines that this change can be made with an ALTER TABLESPACE statement and issues the statement.

Related tasks

"Altering table spaces" on page 452
You can make changes to table spaces or partitions, such as changing the name or lock size or converting a table space to be a partitioned. Depending on the change that you want to make, use either the AL or ALT line command.

Redefining partitions in a partitioned table space that uses index-controlled partitioning

Index-controlled partitioning is a type of partitioning in which partition boundaries for a partitioned table are controlled by values that are specified on the CREATE INDEX statement.

About this task

This procedure assumes that you want to increase the number of partitions. If you want to only update the existing limit key values, navigate to the indexes panel (1.X) and use the ALT function on the associated partitioning index. See "Altering indexes" on page 490.

Procedure

To redefine partitions in a partitioned table space that uses index-controlled partitioning:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option S, and press Enter.

3. On the **Table Spaces (ADB21S)** panel, enter the ALT line command against the partitioned table space that you want to redefine:

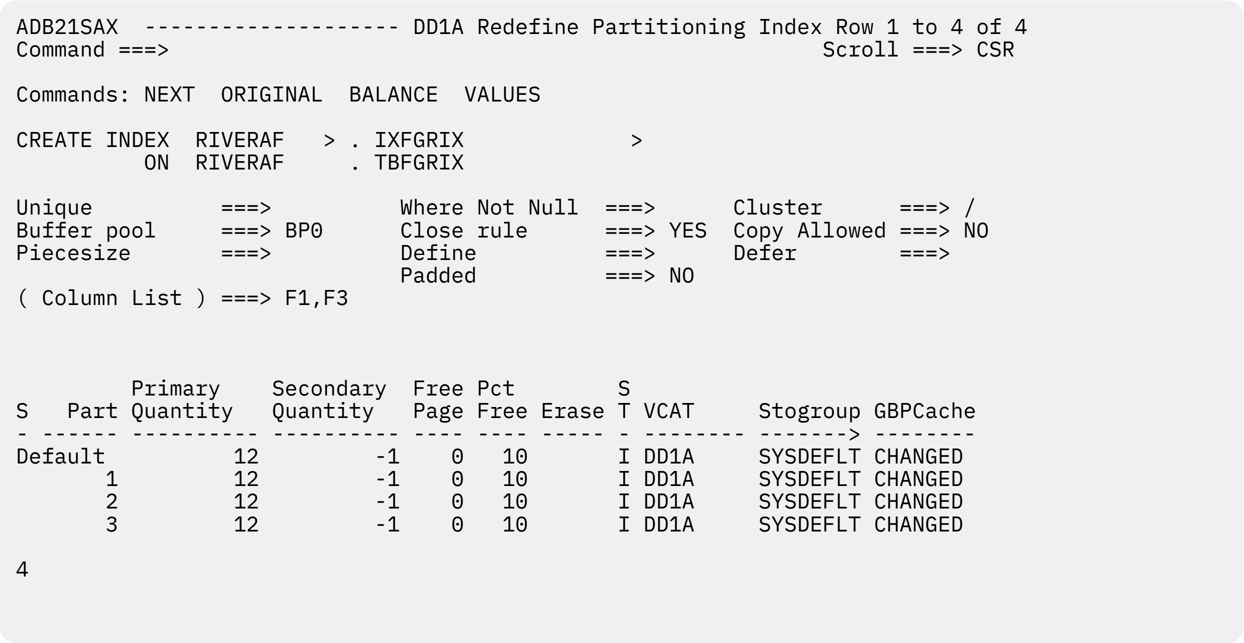
```
ADB21S in ------------------ DD1A Table Spaces --------------- Row 1 to 5 of 5
Command ===>                                                   Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL      DROP  MOVETB
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands
                                                                              C
Select Name     DB Name    Parts Bpool  L E S I C  Tbls   Act pages  Segsz T L O
       *        *             * *      * * * * *     *           *      * * * *
------ -------- -------- ------ ------ - - - - - ----- ---------- ------ - - -
ALT    TSFGRIX  DBFGRIX       3 BP0    A N A N Y     1          -1      0   Y
       TSFGRIX1 DBFGRIX1      3 BP0    A N A N Y     1          -1      0   Y
       TSFGRIX2 DBFGRIX2      4 BP0    P N A N N     1          -1      0   Y
       TSFGRIX3 DBFGRIX3      6 BP1    A N A N N     1          -1      0   Y
       TSFGRIXR DBFGRIXR      3 BP0    A N A N Y     1          -1      0   Y
```

4. If the **Change Management Prompt (ADB2CMRO)** panel is displayed, specify whether you want to use Change Management, and press Enter.
5. On the **Redefine Table Space (ADB21SAR)** panel, increase the value in the **Numparts** field, and press Enter. (For example, change the **Numparts** value from 3 to 4.)

   A new partition row is added to the list of partitions.

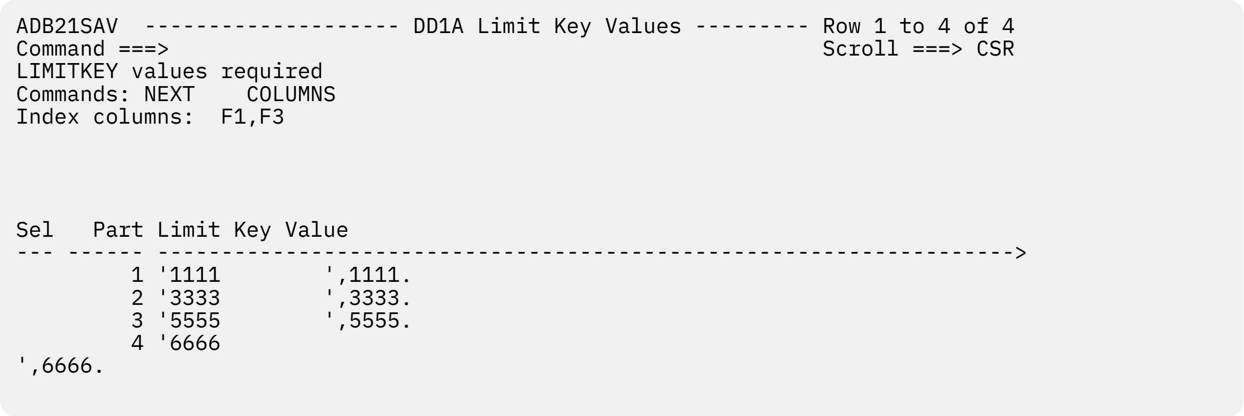
```
ADB21SAR  ------------------ DD1A Redefine Table Space ------- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Commands: NEXT  ORIGINAL  BALANCE  VALUES  MAKEPBG  MAKEPBR  MAKEPBR2
Line commands: S - Split part   R - Remove part  O - Original data
               C - Clear data                    ? - Show all line
commands
CREATE TABLESPACE: TSFGRIX  IN  DBFGRIX

Owner . . . . . . .           > Owner type  . .   (U/R)

Numparts  . . . . . 3                             LOB . . . . . NO
Define  . . . . . . YES                           LOG . . . . . YES
Member Cluster  . . NO         SEGSIZE . . . . 0      CCSID . . . . EBCDIC
Buffer Pool . . . . BP0        Close Rule  . . YES    Max Rows  . . 255
Lock Size . . . . . ANY        Lock Part . . . NO     Lock Max  . . SYSTEM
Max Partitions  . . 0          PAGENUM . . . .        Insert Algo . 0

                                       C E T S
S   Part       Pqty    Sqty  FP PF PFU O R M T VCAT     Stogroup GBPCach DSSIZE
- ------ ---------- ------- --- -- --- - - - - -------- -------> ------- ------
Default:         12      -1   0  5   0 N N Y I DD1A     SYSDEFLT CHANGED
       1
       2
       3
******************************* END OF DB2 DATA *******************************
```

   Note: Insert Algo is displayed only if you are running Db2 12 for z/OS.
6. On the **Redefine Table Space (ADB21SAR)** panel, enter NEXT on the command line.
7. On the **Redefine Partitioning Index (ADB21SAX)** panel, enter NEXT on the command line.



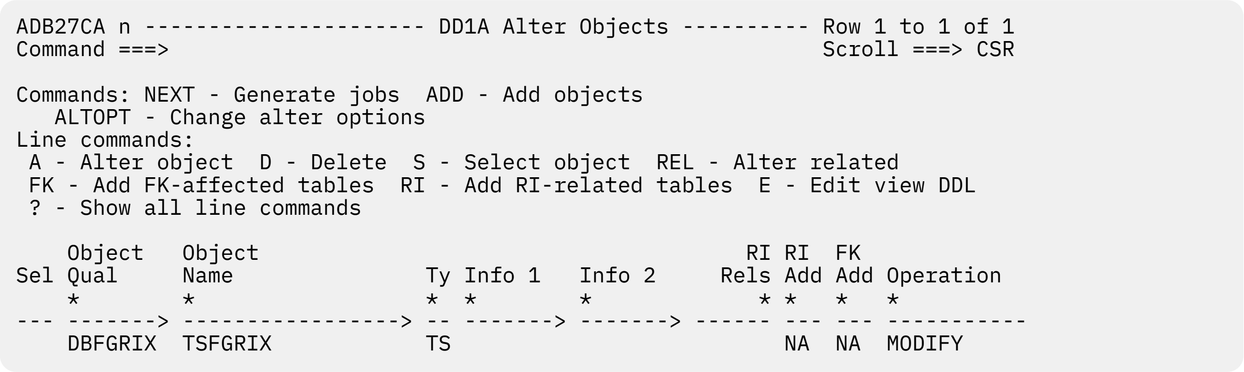
```
ADB21SAX  -------------------- DD1A Redefine Partitioning Index Row 1 to 4 of 4
Command ===>                                                   Scroll ===> CSR

Commands: NEXT  ORIGINAL  BALANCE  VALUES

CREATE INDEX  RIVERAF   > . IXFGRIX             >
          ON  RIVERAF     . TBFGRIX

Unique          ===>          Where Not Null  ===>      Cluster      ===> /
Buffer pool     ===> BP0      Close rule      ===> YES  Copy Allowed ===> NO
Piecesize       ===>          Define          ===>      Defer        ===>
                              Padded          ===> NO
( Column List ) ===> F1,F3



         Primary    Secondary  Free Pct        S
S   Part Quantity   Quantity   Page Free Erase T VCAT     Stogroup GBPCache
- ------ ---------- ---------- ---- ---- ----- - -------- -------> --------
Default          12         -1    0   10       I DD1A     SYSDEFLT CHANGED
       1         12         -1    0   10       I DD1A     SYSDEFLT CHANGED
       2         12         -1    0   10       I DD1A     SYSDEFLT CHANGED
       3         12         -1    0   10       I DD1A     SYSDEFLT CHANGED

4
```

8. On the **Limit Key Values (ADB21SAV)** panel, specify a limit key value for the added partition, and enter NEXT on the command line.



```
ADB21SAV  -------------------- DD1A Limit Key Values --------- Row 1 to 4 of 4
Command ===>                                                   Scroll ===> CSR
LIMITKEY values required
Commands: NEXT    COLUMNS
Index columns:  F1,F3




Sel   Part Limit Key Value
--- ------ ------------------------------------------------------------------->
         1 '1111        ',1111.
         2 '3333        ',3333.
         3 '5555        ',5555.
         4 '6666
',6666.
```

9. On the **Alter Objects (ADB27CA)** panel, enter NEXT on the command line:



```
ADB27CA n --------------------- DD1A Alter Objects ---------- Row 1 to 1 of 1
Command ===>                                                   Scroll ===> CSR

Commands: NEXT - Generate jobs  ADD - Add objects
   ALTOPT - Change alter options
Line commands:
 A - Alter object  D - Delete  S - Select object  REL - Alter related
 FK - Add FK-affected tables  RI - Add RI-related tables  E - Edit view DDL
 ? - Show all line commands

    Object   Object                                  RI RI  FK
Sel Qual     Name               Ty Info 1   Info 2     Rels Add Add Operation
    *        *                  *  *        *            * *   *   *
--- -------> -----------------> -- -------> -------> ------ --- --- -----------
    DBFGRIX  TSFGRIX            TS                          NA  NA  MODIFY
```

10. Generate an ALT job.

Related tasks

“Redefining partitions in a partitioned table space that uses table-controlled partitioning” on page 459
Table-controlled partitioning does not require an index for partitioning and is defined by PARTITION clauses on the CREATE TABLE statement.

“Altering table spaces” on page 452

You can make changes to table spaces or partitions, such as changing the name or lock size or converting a table space to be a partitioned. Depending on the change that you want to make, use either the AL or ALT line command.

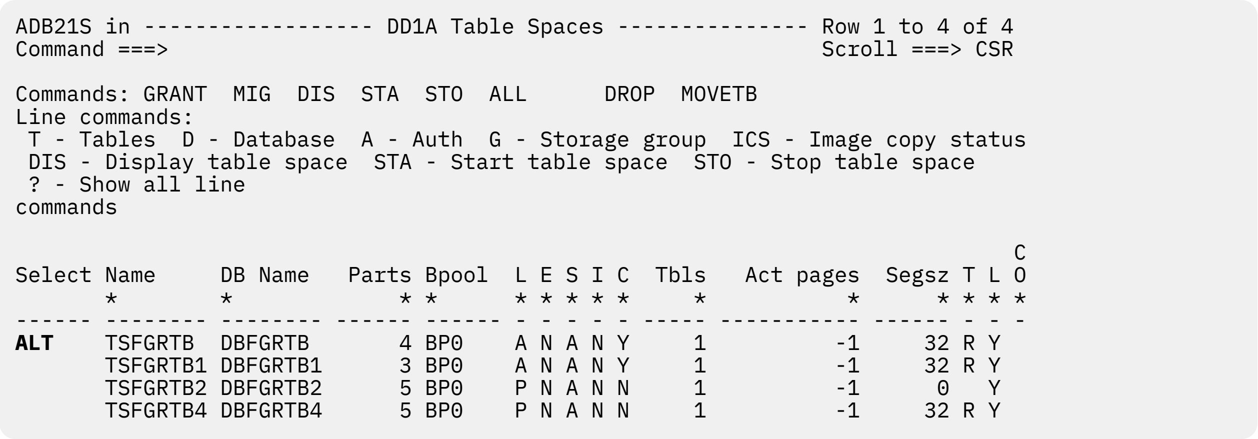
Redefining partitions in a partitioned table space that uses table-controlled partitioning

Table-controlled partitioning does not require an index for partitioning and is defined by PARTITION clauses on the CREATE TABLE statement.

Procedure

To redefine partitions in a partitioned table space that uses table-controlled partitioning:

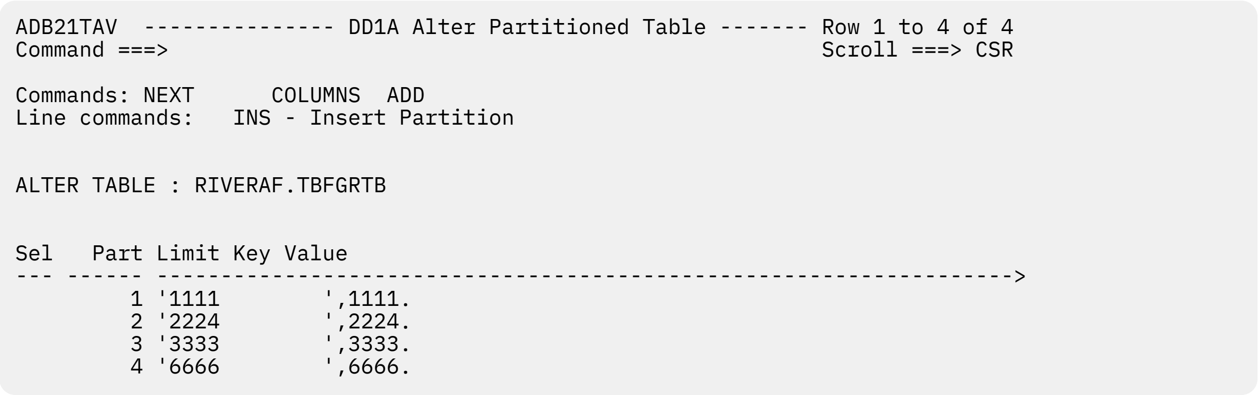
1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option S, and press Enter.
3. On the **Table Spaces (ADB21S)** panel, enter the ALT line command against the partitioned table space that you want to redefine:



```
ADB21S in ------------------ DD1A Table Spaces --------------- Row 1 to 4 of 4
Command ===>                                                   Scroll ===> CSR

Commands: GRANT  MIG  DIS  STA  STO  ALL      DROP  MOVETB
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line
commands

                                                                              C
Select Name     DB Name   Parts Bpool  L E S I C  Tbls   Act pages  Segsz T L O
       *        *             * *      * * * * *     *           *      * * * *
------ -------- -------- ------ ------ - - - - - ----- ----------- ------ - - -
ALT    TSFGRTB  DBFGRTB       4 BP0    A N A N Y     1          -1     32 R Y
       TSFGRTB1 DBFGRTB1      3 BP0    A N A N Y     1          -1     32 R Y
       TSFGRTB2 DBFGRTB2      5 BP0    P N A N N     1          -1      0   Y
       TSFGRTB4 DBFGRTB4      5 BP0    P N A N N     1          -1     32 R Y
```

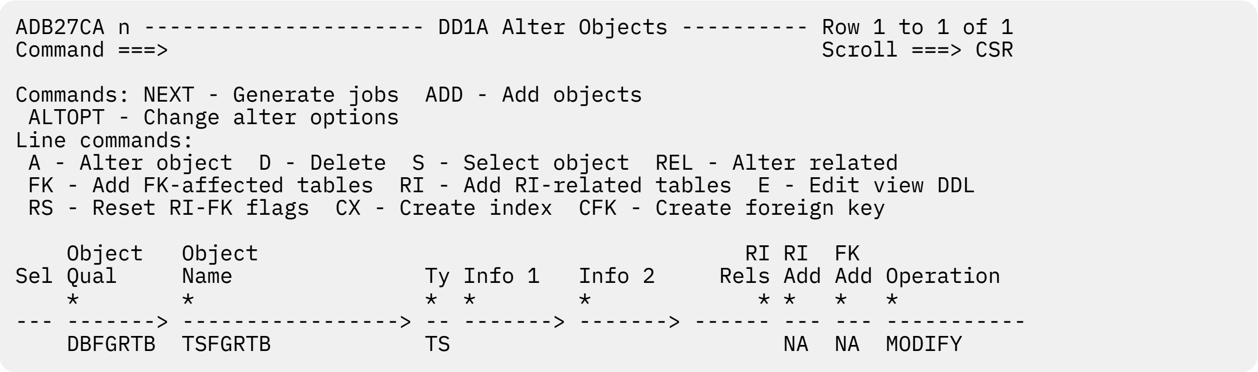
4. If the **Change Management Prompt (ADB2CMRO)** panel is displayed, specify whether you want to use Change Management, and press Enter.
5. On the **Redefine Table Space (ADB21SAR)** panel, enter VALUES on the command line.

   If VALUES is not listed as a command, the object is not a partitioned table space.
6. On the **Alter Partitioned Table (ADB21TAV)** panel, edit the limit key value that you want to update and then enter NEXT on the command line.



```
ADB21TAV  --------------- DD1A Alter Partitioned Table ------- Row 1 to 4 of 4
Command ===>                                                   Scroll ===> CSR

Commands: NEXT      COLUMNS  ADD
Line commands:   INS - Insert Partition


ALTER TABLE : RIVERAF.TBFGRTB


Sel   Part Limit Key Value
--- ------ -------------------------------------------------------------------->
         1 '1111        ',1111.
         2 '2224        ',2224.
         3 '3333        ',3333.
         4 '6666        ',6666.
```

7. On the **Redefine Table Space (ADB21SAR)** panel, enter NEXT on the command line.
8. On the **Alter Objects (ADB27CA)** panel, enter NEXT on the command line:



```
ADB27CA n --------------------- DD1A Alter Objects ---------- Row 1 to 1 of 1
Command ===>                                                   Scroll ===> CSR

Commands: NEXT - Generate jobs  ADD - Add objects
 ALTOPT - Change alter options
Line commands:
 A - Alter object  D - Delete  S - Select object  REL - Alter related
 FK - Add FK-affected tables  RI - Add RI-related tables  E - Edit view DDL
 RS - Reset RI-FK flags  CX - Create index  CFK - Create foreign key

    Object   Object                                      RI RI  FK
Sel Qual     Name               Ty Info 1   Info 2     Rels Add Add Operation
    *        *                  *  *        *             * *   *   *
--- -------> -----------------> -- -------> -------> ------ --- --- -----------
    DBFGRTB  TSFGRTB            TS                          NA  NA  MODIFY
```

9. Generate an ALT job.

Related tasks

“Redefining partitions in a partitioned table space that uses index-controlled partitioning” on page 456
Index-controlled partitioning is a type of partitioning in which partition boundaries for a partitioned table are controlled by values that are specified on the CREATE INDEX statement.

“Altering table spaces” on page 452
You can make changes to table spaces or partitions, such as changing the name or lock size or converting a table space to be a partitioned. Depending on the change that you want to make, use either the AL or ALT line command.

Altering tables

You can make changes to tables, such as adding columns, changing the name, or dropping constraints. Depending on the change that you want to make, use either the AL or ALT line command.

About this task

You can use Db2 Admin Tool to make the following changes to a table:

- Change the database, table space, owner, or name of a table
- Modify the definitions of table columns (with some restrictions)
- Change the sequence of the columns in a table
- Drop columns
- Insert new columns
- Drop or add unique, check, and foreign key constraints
- Modify table attributes such as auditing, data capture, validation procedure, restrict on drop, index access, and append processing
- Modify the data organization of the table
- Activate or deactivate row and column access control
- Drop or add column masks
- Add system or business time periods
- Drop or add versioning
- Add or alter partitions
- Add partitioning keys
- Drop or add clone tables

Restrictions:

- Changes to column names are retrofitted to views. All other column actions are not retrofitted, and any changes to a column's data type are not verified against the views.
- All columns that comprise the partitioning columns of the table cannot be dropped.

- A warning is displayed if you attempt to modify columns in the primary key. With the UP line command (update primary key), you can circumvent the warning. You can use the ADDFK primary command to propagate the primary key update to foreign-key related tables.
- If you modify columns that are in a foreign key, Db2 Admin Tool does not automatically modify the primary key of the parent tables. To propagate the column updates to primary and foreign keys, use the ADD primary command from the **ALTER Table (ADB27C)** panel to initiate the Alter Tables dialog, where RI-related tables or other tables can be included in the ALTER JCL stream.
- Db2 Admin Tool informs you when a specific data type conversion is allowed. See “Db2 Admin Tool data type conversions” on page 249.
- If you modify a table that has a security label column or LOB columns or if you are creating a work statement list, you cannot specify HPU in the **Unload Method** field on the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel. For work statement lists, you must specify UNLOAD.
- The HPU PARMLIB parameter must be set to the default value.

Use the AL line command to make changes that are supported by the ALTER TABLE statement. For example, you can add a primary key or partitioning key, add or insert a partition, alter a partition, rotate a partition, or drop a column. These changes are called non-intrusive changes. A *non-intrusive alter* is one in which the table does not have to be dropped and re-created. When you use the AL line command, an ALTER statement is generated.

Use the ALT line command to make table changes that you cannot make with an ALTER statement, such as renaming the table or adding a partition. In this case, the table is dropped and recreated and said to be *redefined*. The ALT line command can also make some, but not all, changes that are supported by ALTER statements. Additionally, ALT can process multiple objects and run utilities.

- “Altering a table by using the AL line command” on page 461
- “Altering a table by using the ALT line command” on page 461

Altering a table by using the AL line command

Procedure

To alter a table by using the AL line command:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the AL line command against the table that you want to alter.
4. On the **Alter Table (ADB21TA)** panel, type S next to the change that you want to select it, and press Enter.
5. Complete the subsequent panels, and press Enter to run the ALTER TABLE statement.

Related tasks

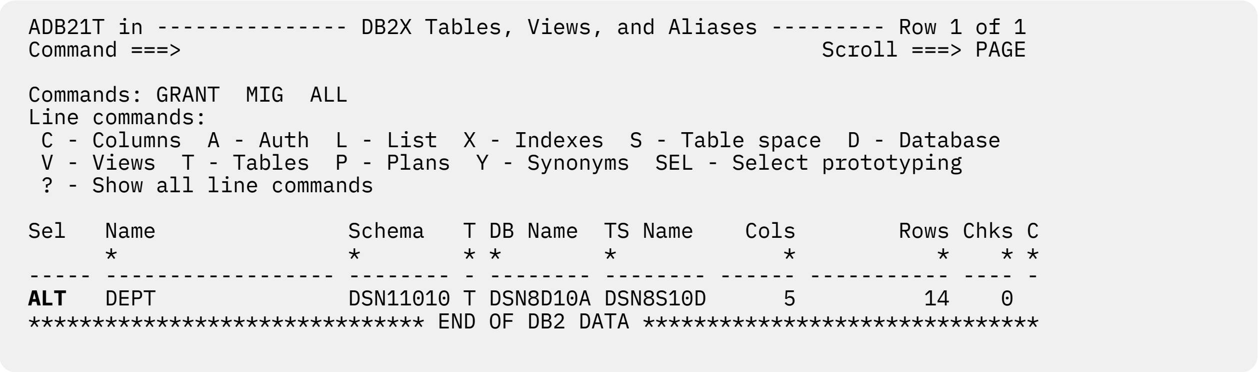
“Examples of altering a table by using the AL line command” on page 463

Altering a table by using the ALT line command

Procedure

To alter a table by using the ALT line command:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT line command against the table to be changed, and press Enter:

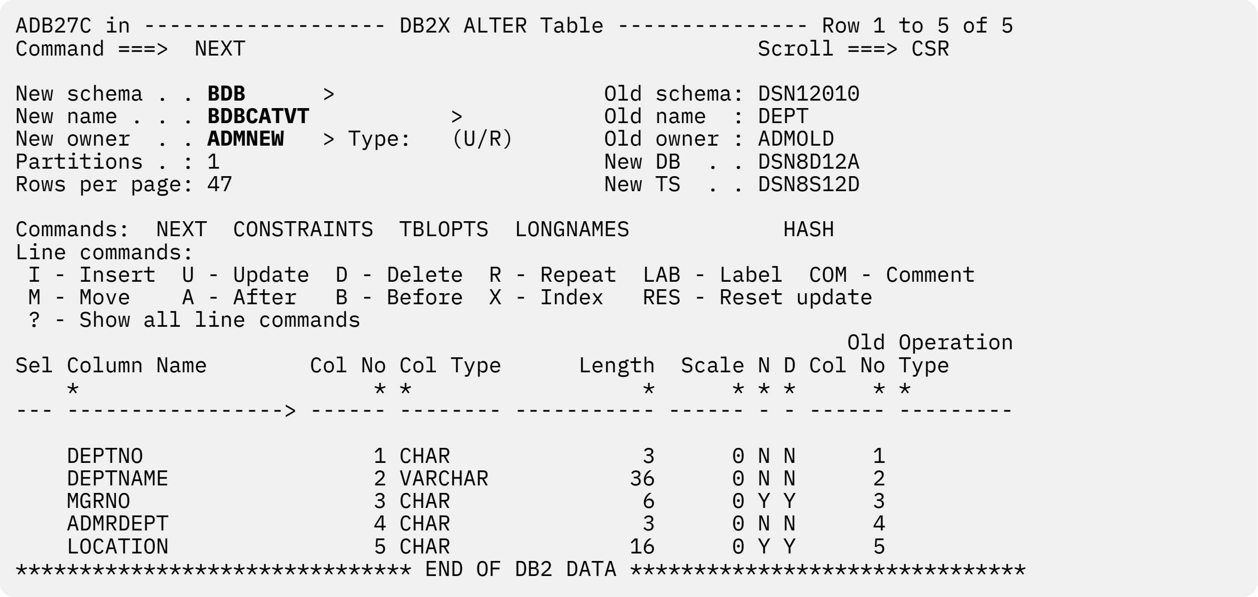


```
ADB21T in --------------- DB2X Tables, Views, and Aliases --------- Row 1 of 1
Command ===>                                                  Scroll ===> PAGE

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

Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C
      *                  *        * *        *             *           *    * *
----- ------------------ -------- - -------- -------- ------ ----------- ---- -
ALT   DEPT               DSN11010 T DSN8D10A DSN8S10D      5          14    0
******************************* END OF DB2 DATA *******************************
```

Figure 246. Tables, Views, and Aliases panel (ADB21T)

4. On the **ALTER Table (ADB27C)** panel, specify any new attribute values.

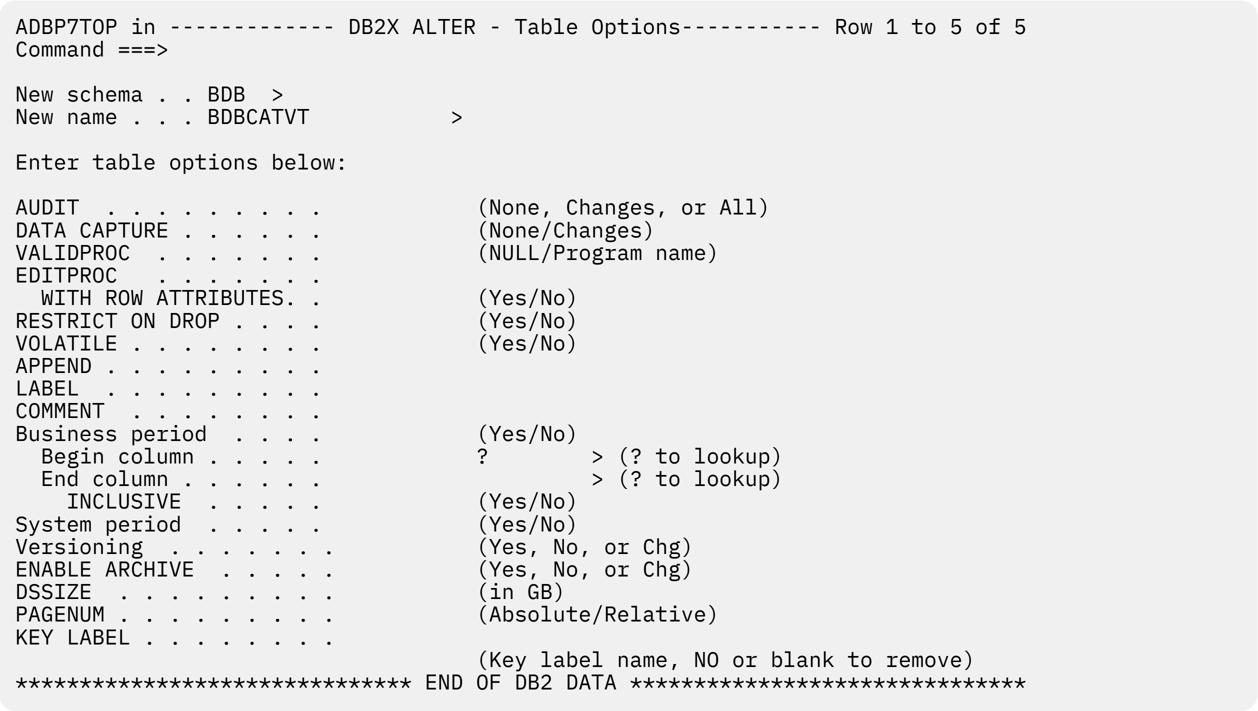
   In the following example, the schema, name, and owner are changed:



```
ADB27C in ------------------- DB2X ALTER Table --------------- Row 1 to 5 of 5
Command ===>  NEXT                                        Scroll ===> CSR

New schema . . BDB      >                     Old schema: DSN12010
New name . . . BDBCATVT          >            Old name  : DEPT
New owner  . . ADMNEW   > Type:   (U/R)       Old owner : ADMOLD
Partitions . : 1                              New DB  . . DSN8D12A
Rows per page: 47                             New TS  . . DSN8S12D

Commands:  NEXT  CONSTRAINTS  TBLOPTS  LONGNAMES            HASH
Line commands:
 I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index   RES - Reset update
 ? - Show all line commands
                                                                 Old Operation
Sel Column Name        Col No Col Type      Length  Scale N D Col No Type
    *                       * *                  *      * * *      * *
--- ----------------> ------ -------- ---------- ------ - - ------ ---------

    DEPTNO                 1 CHAR              3      0 N N      1
    DEPTNAME               2 VARCHAR          36      0 N N      2
    MGRNO                  3 CHAR              6      0 Y Y      3
    ADMRDEPT               4 CHAR              3      0 N N      4
    LOCATION               5 CHAR             16      0 Y Y      5
******************************* END OF DB2 DATA ******************************
```

*Figure 247. **ALTER Table (ADB27C)** panel*

5. Optional: To change additional attributes, such as a period definition for the table, issue the TBLOPTS command, and press Enter. Then, on the **Alter - Table Options (ADBP7TOP)** panel, specify any additional changes to the table, and press Enter:



```
ADBP7TOP in ------------- DB2X ALTER - Table Options----------- Row 1 to 5 of 5
Command ===>

New schema . . BDB  >
New name . . . BDBCATVT           >

Enter table options below:

AUDIT  . . . . . . . . .            (None, Changes, or All)
DATA CAPTURE . . . . . .            (None/Changes)
VALIDPROC  . . . . . . .            (NULL/Program name)
EDITPROC   . . . . . . .
  WITH ROW ATTRIBUTES. .            (Yes/No)
RESTRICT ON DROP . . . .            (Yes/No)
VOLATILE . . . . . . . .            (Yes/No)
APPEND . . . . . . . . .
LABEL  . . . . . . . . .
COMMENT  . . . . . . . .
Business period  . . . .            (Yes/No)
  Begin column . . . . .            ?        > (? to lookup)
  End column . . . . . .                     > (? to lookup)
    INCLUSIVE  . . . . .            (Yes/No)
System period  . . . . .            (Yes/No)
Versioning  . . . . . . .           (Yes, No, or Chg)
ENABLE ARCHIVE  . . . . .           (Yes, No, or Chg)
DSSIZE  . . . . . . . . .           (in GB)
PAGENUM . . . . . . . . .           (Absolute/Relative)
KEY LABEL . . . . . . . .
                                    (Key label name, NO or blank to remove)
******************************* END OF DB2 DATA *******************************
```

*Figure 248. **Alter - Table Options (ADBP7TOP)** panel*

Tip: For changes to some of these attributes, such as **KEY LABEL**, you must run the REORG utility to materialize the change. See Pending data definition changes (Db2 12 for z/OS).

6. On the **ALTER Table (ADB27C)** panel, issue the NEXT command, and press Enter.
7. Generate an ALT job.

Related tasks

"Examples of redefining a table by using the ALT line command" on page 470

Examples of altering a table by using the AL line command

You can use the AL line command to make changes to your tables that are supported by the ALTER TABLE statement.

- "Adding a primary key to the table" on page 463
- "Adding a partitioning key to the table" on page 464
- "Adding a partition to the table " on page 464
- "Inserting a partition into a table " on page 465
- "Altering a partition" on page 466
- "Rotating a partition" on page 467
- "Dropping a column" on page 467
- "Adding or changing a key label" on page 469

Adding a primary key to the table

Procedure

To add a primary key to the table:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option T, and press Enter.

3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the AL line command against the table that you want to alter.
4. On the **Alter Table (ADB21TA)** panel, type an S before **ADD PRIMARY KEY**, and press Enter.
5. On the **Add Primary Key Constraint (ADB21TAN)** panel, in the **PRIMARY KEY** field, specify the column names.

   For help selecting the columns for the primary key, use the COLUMNS primary command to display a list of the columns. Then, use the *nn* (Sequence) line command to specify a number for the relative position of each column that you want to include in the primary key. When you are done, press PF3 to return to the previous panel.
6. Optional: In the **Constraint** field, specify a name for the primary key constraint.
7. Press Enter to run the ALTER TABLE statement that adds the primary key.

Adding a partitioning key to the table

Procedure

To add a partitioning key to the table:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the AL line command against the table that you want to alter.
4. On the **Alter Table (ADB21TA)** panel, type an S before **ADD PARTITIONING KEY**, and press Enter.
5. On the **Alter Table (ADB21TAP)** panel, select the columns to be part of the partitioning key by using the Sequence & order line command.

   For example, specifying 1D next to a column name assigns that column to be first in the partitioning key with the values in descending order.

   To clear all of your changes, use the ORIGINAL primary command.
6. Issue the NEXT primary command.
7. On the **Alter Partitioned Table (ADB21TAV)** panel, enter the limit key values.

   For help determining the limit key values, you can use the COLUMNS primary command to list the details of the columns that are part of the key.
8. Enter the NEXT primary command to run the ALTER TABLE statement that adds the partitioning key.

Adding a partition to the table

Procedure

To add a partition to the table:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the AL line command against the table that you want to alter.
4. On the **Alter Table (ADB21TA)** panel, type an S before **ADD/INSERT/ALTER PARTITION** and press Enter.

   (This option is displayed only if applicable. The INSERT option is displayed only if you are running Db2 12 for z/OS.)
5. On the **Alter Partitioned Table (ADB21TAV)** panel, issue the ADD primary command to add a row with the next partition number.
6. Enter the limit key value for the new partition.

7. Issue the NEXT primary command to run the ALTER TABLE statement to add the partition.

   If the **Statement Execution Prompt (ADB2PSTM)** panel is displayed, enter the appropriate action to execute the statement.
8. On the **Alter Table - Utilities (ADB21TAU)** panel, specify any utilities that you want to run, and press Enter.
9. Follow the instructions on any subsequent utility panels.

   If you requested any utility operations, a utility job is generated.
10. If a utility job is generated, submit that job to run the requested utilities.

Inserting a partition into a table

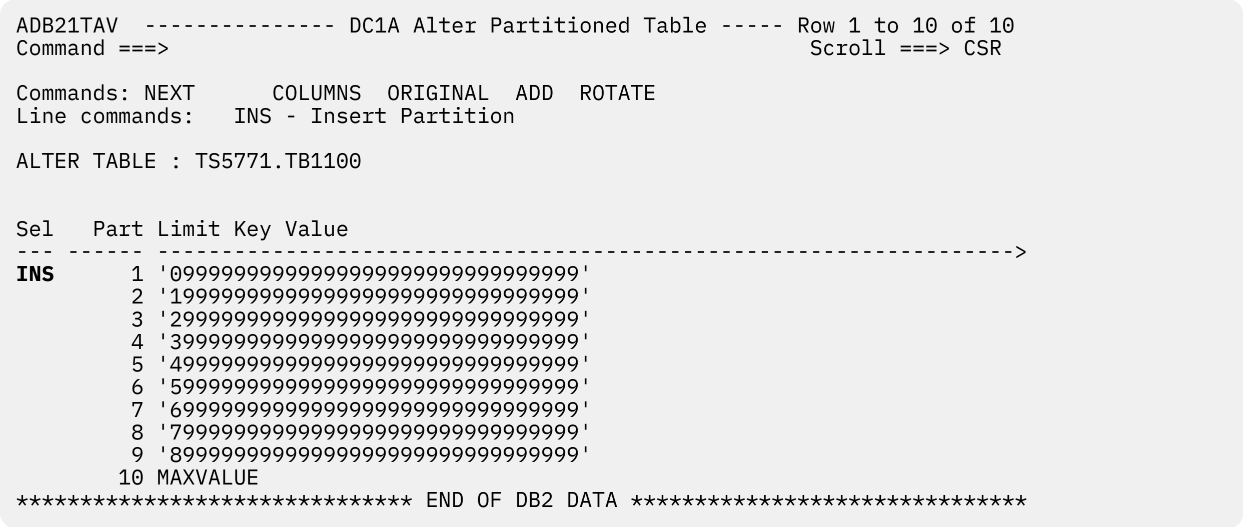
Procedure

To insert a partition into a table:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the AL line command against the table that you want to alter.
4. On the **Alter Table (ADB21TA)** panel, type an S before **ADD/INSERT/ALTER PARTITION** and press Enter.

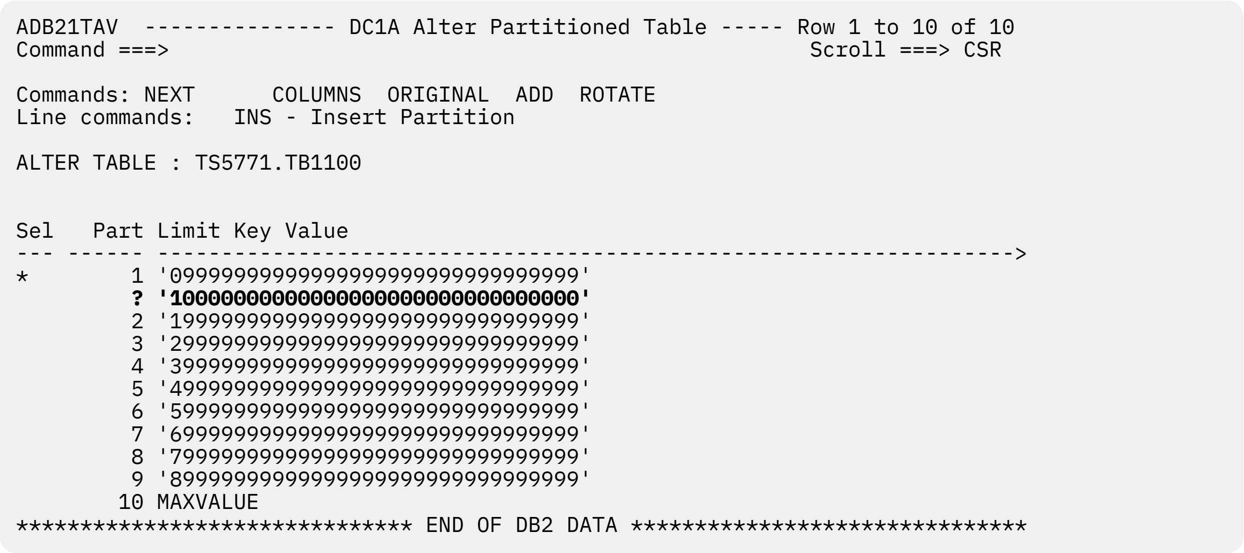
   (This option is displayed only if applicable. The INSERT option is displayed only if you are running Db2 12 for z/OS.)
5. On the **Alter Partitioned Table (ADB21TAV)** panel, issue the INS line command to insert a new partition, as shown in the following example:

   Note: You do not have to issue the INS line command next to the row where you want to insert the partition. The logical partition number will be determined by the limit key, not by where you specify INS. For example, if you want to insert a partition before partition 1, you can issue the INS line command next to any row and, as part of the next step, specify the lowest limit key.



```
ADB21TAV  --------------- DC1A Alter Partitioned Table ----- Row 1 to 10 of 10
Command ===>                                                  Scroll ===> CSR

Commands: NEXT      COLUMNS  ORIGINAL  ADD  ROTATE
Line commands:   INS - Insert Partition

ALTER TABLE : TS5771.TB1100


Sel   Part Limit Key Value
--- ------ ------------------------------------------------------------------->
INS      1 '09999999999999999999999999999999'
         2 '19999999999999999999999999999999'
         3 '29999999999999999999999999999999'
         4 '39999999999999999999999999999999'
         5 '49999999999999999999999999999999'
         6 '59999999999999999999999999999999'
         7 '69999999999999999999999999999999'
         8 '79999999999999999999999999999999'
         9 '89999999999999999999999999999999'
        10 MAXVALUE
******************************* END OF DB2 DATA *******************************
```

6. Enter the limit key value for the new partition, as shown in the following example:



```
ADB21TAV  --------------- DC1A Alter Partitioned Table ----- Row 1 to 10 of 10
Command ===>                                                  Scroll ===> CSR

Commands: NEXT      COLUMNS  ORIGINAL  ADD  ROTATE
Line commands:   INS - Insert Partition

ALTER TABLE : TS5771.TB1100


Sel   Part Limit Key Value
--- ------ -------------------------------------------------------------------->
*        1 '09999999999999999999999999999999'
         ? '10000000000000000000000000000000'
         2 '19999999999999999999999999999999'
         3 '29999999999999999999999999999999'
         4 '39999999999999999999999999999999'
         5 '49999999999999999999999999999999'
         6 '59999999999999999999999999999999'
         7 '69999999999999999999999999999999'
         8 '79999999999999999999999999999999'
         9 '89999999999999999999999999999999'
        10 MAXVALUE
******************************* END OF DB2 DATA ********************************
```

The logical partition number of the inserted partition is determined by Db2 after the ALTER statement is executed. Db2 determines the logical partition number based on the values of the specified limit key.

7. Issue the NEXT primary command to run the ALTER TABLE statement to add the partition.

   If the **Statement Execution Prompt (ADB2PSTM)** panel is displayed, enter the appropriate action to execute the statement.

   The **Restrictive State (ADBPM610)** panel is displayed to indicate that the table is in advisory REORG-pending status. You must run the REORG TABLESPACE utility to materialize the change that inserts the partition.
8. Specify YES in the **REORG** field and any other options that you want, and press Enter.
9. Follow the instructions on the subsequent panels to generate a REORG job.
10. Submit the REORG job to complete the process of inserting a new partition.

Altering a partition

Procedure

To alter a partition:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the AL line command against the table that you want to alter.
4. On the **Alter Table (ADB21TA)** panel, type an S before **ADD/INSERT/ALTER PARTITION** and press Enter.

   (This option is displayed only if applicable. The INSERT option is displayed only if you are running Db2 12 for z/OS.)
5. On the **Alter Partitioned Table (ADB21TAV)** panel, change the limit key values for any of the partitions.
6. Issue the NEXT primary command to run the ALTER TABLE statement to alter the partitions with their new values.

Rotating a partition

Procedure

To rotate a partition:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the AL line command against the table that you want to alter.
4. On the **Alter Table (ADB21TA)** panel, type an S before **ADD/INSERT/ALTER PARTITION** and press Enter.

   (This option is displayed only if applicable. The INSERT option is displayed only if you are running Db2 12 for z/OS.)
5. On the **Alter Partitioned Table (ADB21TAV)** panel, issue the ROTATE primary command.
6. On the **Alter Table (ADB21TAR)** panel, enter the limit key value for the rotated partition, and press Enter.
7. On the **Alter Partitioned Table (ADB21TAV)** panel, issue the NEXT command.
8. On the **Alter Table - Utilities (ADB21TAU)** panel, specify any utilities that you want to run, and press Enter.

   The ROTATE statement is held until all other ALTER statements are executed. If the first logical partition of the table space is in REORG, run the REORG utility before running ROTATE.
9. Submit the generated job to rotate the partition.

   The generated job contains SQL that is similar to the following example SQL:

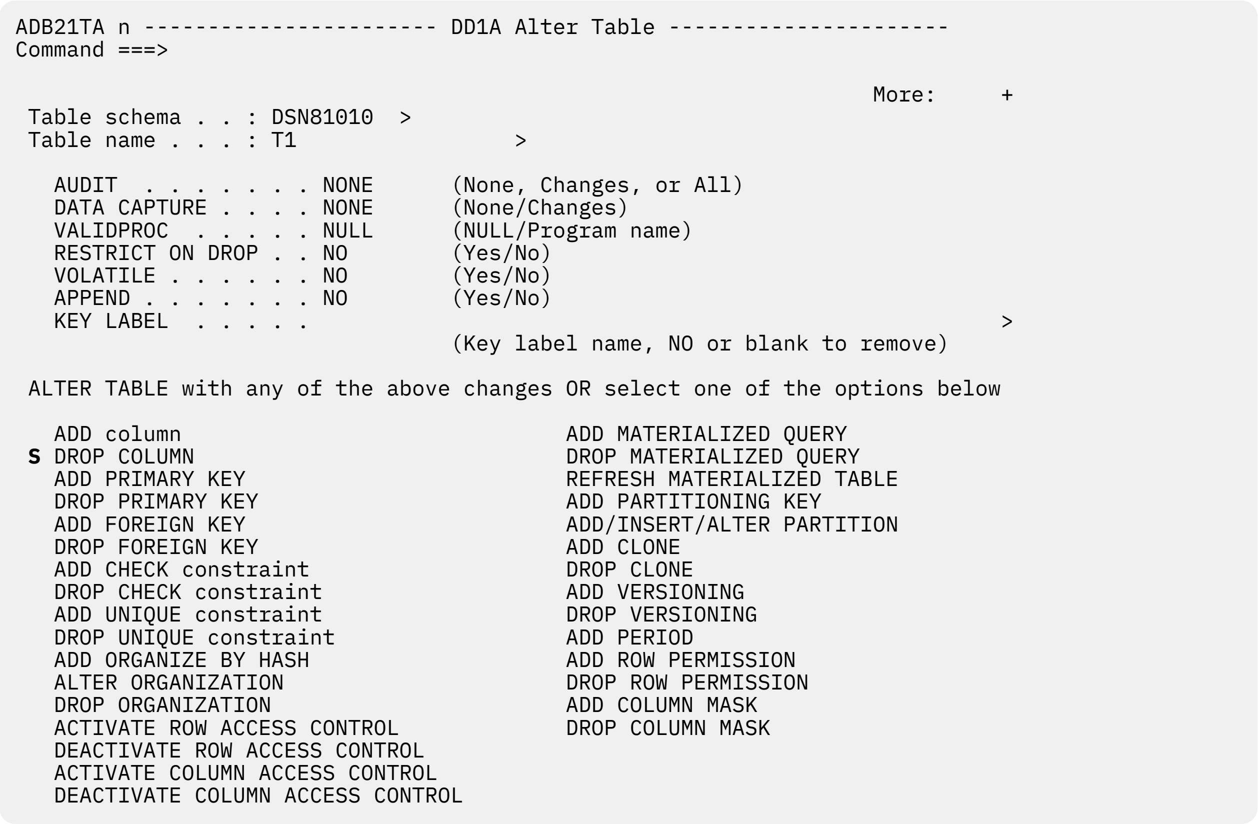
   ```
   ALTER TABLE "SMITHJR"."TBADAJ01" ROTATE PARTITION FIRST TO LAST  ENDING
   AT ('10500') RESET;
   ```

Dropping a column

Procedure

To drop a column:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the AL line command against the table that you want to alter.
4. On the **Alter Table (ADB21TA)** panel, type an S before **DROP COLUMN** and press Enter:

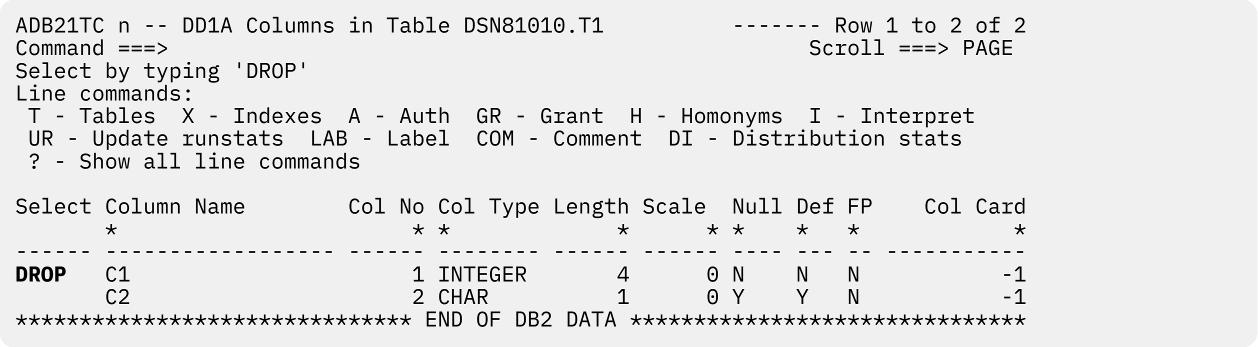


```
ADB21TA n ----------------------- DD1A Alter Table ----------------------
Command ===>
                                                                    More:     +
 Table schema . . : DSN81010  >
 Table name . . . : T1                 >

   AUDIT  . . . . . . . NONE      (None, Changes, or All)
   DATA CAPTURE . . . . NONE      (None/Changes)
   VALIDPROC  . . . . . NULL      (NULL/Program name)
   RESTRICT ON DROP . . NO        (Yes/No)
   VOLATILE . . . . . . NO        (Yes/No)
   APPEND . . . . . . . NO        (Yes/No)
   KEY LABEL  . . . . .                                                   >
                                  (Key label name, NO or blank to remove)

 ALTER TABLE with any of the above changes OR select one of the options below

   ADD column                              ADD MATERIALIZED QUERY
 S DROP COLUMN                             DROP MATERIALIZED QUERY
   ADD PRIMARY KEY                         REFRESH MATERIALIZED TABLE
   DROP PRIMARY KEY                        ADD PARTITIONING KEY
   ADD FOREIGN KEY                         ADD/INSERT/ALTER PARTITION
   DROP FOREIGN KEY                        ADD CLONE
   ADD CHECK constraint                    DROP CLONE
   DROP CHECK constraint                   ADD VERSIONING
   ADD UNIQUE constraint                   DROP VERSIONING
   DROP UNIQUE constraint                  ADD PERIOD
   ADD ORGANIZE BY HASH                    ADD ROW PERMISSION
   ALTER ORGANIZATION                      DROP ROW PERMISSION
   DROP ORGANIZATION                       ADD COLUMN MASK
   ACTIVATE ROW ACCESS CONTROL             DROP COLUMN MASK
   DEACTIVATE ROW ACCESS CONTROL
   ACTIVATE COLUMN ACCESS CONTROL
   DEACTIVATE COLUMN ACCESS CONTROL
```

Restriction: DROP COLUMN is not valid if any of the following conditions are true:

- The table in not contained in a universal table space (UTS).
- The table is a materialized query table (MQT).
- The table is referenced in a MQT definition.
- The table contains an edit procedure or a validation-exit procedure.
- The table is in an incomplete state.
- The table is a system-period temporal table.
- The table contains extended indexes that are dependent on the table.
- The table contains triggers that are dependent on the table.
- The table contains row permissions that are dependent on the table.
- The table contains column masks that are dependent on the table.
- The table contains check constraints that are dependent on the table.

5. On the **Columns in Table (ADB21TC)** panel, issue the DROP line command against the column that you want to drop:



```
ADB21TC n -- DD1A Columns in Table DSN81010.T1          ------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE
Select by typing 'DROP'
Line commands:
 T - Tables  X - Indexes  A - Auth  GR - Grant  H - Homonyms  I - Interpret
 UR - Update runstats  LAB - Label  COM - Comment  DI - Distribution stats
 ? - Show all line commands

Select Column Name        Col No Col Type Length Scale  Null Def FP    Col Card
       *                       * *             *      * *    *   *            *
------ ------------------ ------ -------- ------ ------ ---- --- -- -----------
DROP   C1                      1 INTEGER       4      0 N    N   N           -1
       C2                      2 CHAR          1      0 Y    Y   N           -1
******************************* END OF DB2 DATA *******************************
```

Restriction: The DROP line command can be issued against only one column at a time.

Adding or changing a key label

Before you begin

To use key labels, you must be running Db2 12 for z/OS function level 502 or higher.

Procedure

To add or change a key label:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, enter any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the AL line command against the table that you want to alter.
4. On the **Alter Table (ADB21TA)** panel, specify the key label name in the **KEY LABEL** field:

```
ADB21TA n ----------------------- DD1A Alter Table ---------------------- 11:20
 Command ===>


  Table schema . . : TS5816    >
  Table name . . . : TESTKEY             >

    AUDIT  . . . . . . . NONE       (None, Changes, or All)
    DATA CAPTURE . . . . NONE       (None/Changes)
    VALIDPROC  . . . . . NULL       (NULL/Program name)
    RESTRICT ON DROP . . NO         (Yes/No)
    VOLATILE . . . . . . NO         (Yes/No)
    APPEND . . . . . . . NO         (Yes/No)
    KEY LABEL  . . . . .                                                      >
                                    (Key label name, NO or blank to remove)

  ALTER TABLE with any of the above changes OR select one of the options below

    ADD COLUMN                              ADD MATERIALIZED QUERY
    DROP COLUMN                             DROP MATERIALIZED QUERY
    ADD PRIMARY KEY                         REFRESH MATERIALIZED TABLE
    DROP PRIMARY KEY                        ADD PARTITIONING KEY
    ADD FOREIGN KEY                         ADD PARTITION
    DROP FOREIGN KEY                        ADD CLONE
    ADD CHECK constraint                    DROP CLONE
    DROP CHECK constraint                   ADD VERSIONING
    ADD UNIQUE constraint                   DROP VERSIONING
    DROP UNIQUE constraint                  ADD PERIOD
    ADD ORGANIZE BY HASH                    ADD ROW PERMISSION
    ALTER ORGANIZATION                      DROP ROW PERMISSION
    DROP ORGANIZATION                       ADD COLUMN MASK
    ACTIVATE ROW ACCESS CONTROL             DROP COLUMN MASK
    DEACTIVATE ROW ACCESS CONTROL           ENABLE ARCHIVE
    ACTIVATE COLUMN ACCESS CONTROL          DISABLE ARCHIVE
    DEACTIVATE COLUMN ACCESS CONTROL
```

   Note: If the **KEY LABEL** field is not displayed, you are not running Db2 12 for z/OS function level 502 or higher.

   If this field is initially blank, a key label is not defined on this table. You can add a key label by specifying a valid key label name.

   If this field is initially populated, a key label is already defined on the table. You can change the key label by specifying a new value. Alternatively, you can remove it by specifying NO or blank, in which case the NO KEY LABEL clause is added to the ALTER TABLE statement.
5. Press Enter to run the ALTER TABLE statement with the new key label value.
6. To materialize this key label change and encrypt the data set, run the REORG TABLESPACE utility on the table space that contains the table.

Related tasks

“Running Db2 utilities on table spaces ” on page 594

Many Db2 utilities run against table spaces. You can use Db2 Admin Tool to specify the utility options and generate the JCL to run these utilities.

Related reference

ALTER TABLE (Db2 12 for z/OS)

Related information

Using z/OS DFSMS data set encryption to encrypt the data sets associated with a particular table (Db2 12 for z/OS)

Examples of redefining a table by using the ALT line command

You can use the ALT line command to make changes to your tables. With ALT, you can make both changes that are done by ALTER TABLE statements and changes that cannot be done by ALTER TABLE statements. Because ALT supports more than just ALTER statement changes, the ALT process is referred to as *redefining* a table.

- "Inserting a column" on page 470
- "Updating a column" on page 472
- "Adding a unique key to a table" on page 474
- "Changing a unique key" on page 474
- "Adding a column to a primary key" on page 475
- "Renaming a table" on page 477
- "Adding a partition to a table" on page 477
- "Adding a partition to a table in a partition-by-growth (PBG) table space" on page 478
- "Inserting a partition into a table" on page 478
- "Adding a foreign key" on page 479

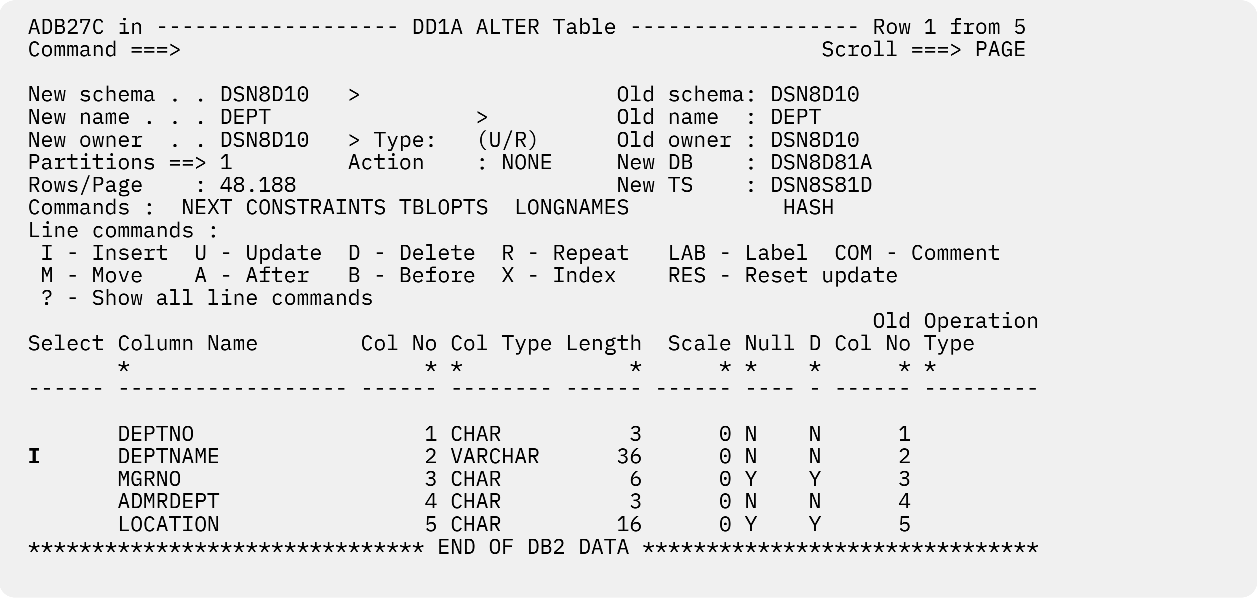
Identity columns: If you are redefining a table that contains an identity column and the table is dropped and re-created, the column definition becomes GENERATED BY DEFAULT to preserve current data values. The first value that is generated for the identity column (specified in the START WITH clause) is also changed. The new START WITH value, which is the value that is to assigned next to the identity column, is the last unassigned value (MAXASSIGNEDVAL in SYSIBM.SYSSEQUENCES) plus the increment value (INCREMENT in SYSIBM.SYSSEQUENCES). If values were cached, any existing unassigned values in the cache that have not been used are lost. Loss of unassigned cached values causes a gap between the last assigned value of the identity column and the new starting value.

Inserting a column

Procedure

To insert a column:

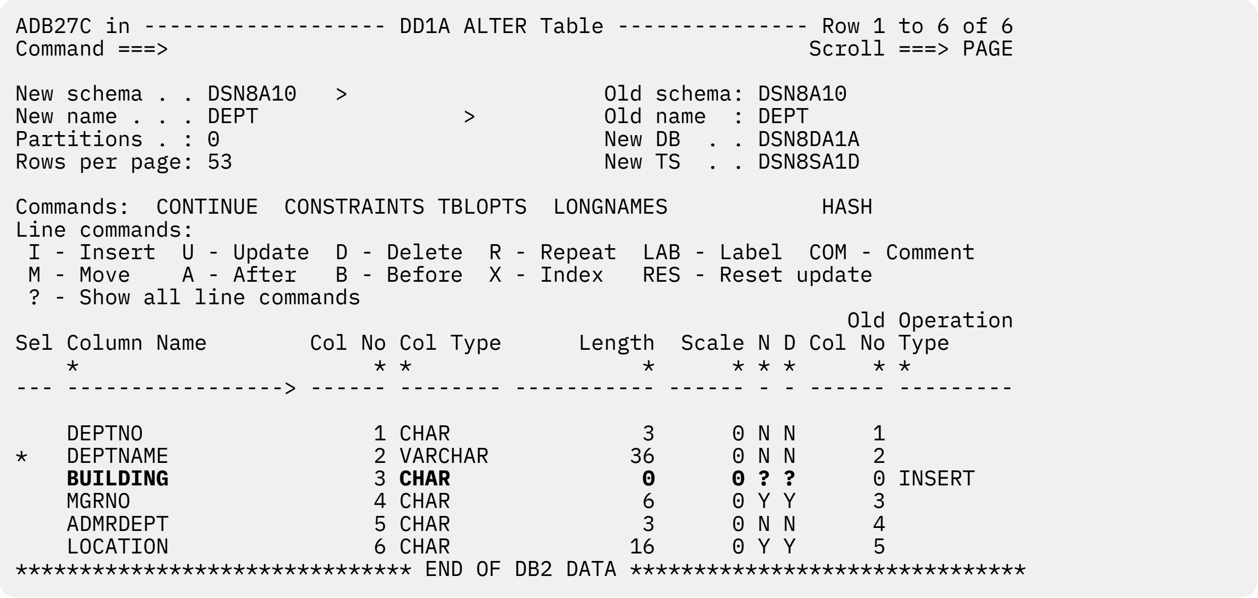
1. On the **DB2 Administration Menu (ADB2)** panel, select option 1.
2. On the **System Catalog (ADB21)** panel, select option T.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT line command against the table to be changed.
4. On the **ALTER Table (ADB27C)** panel, issue the I line command against the column after which you want to insert a new column:



```
ADB27C in ------------------- DD1A ALTER Table ------------------ Row 1 from 5
Command ===>                                                  Scroll ===> PAGE

New schema . . DSN8D10   >                    Old schema: DSN8D10
New name . . . DEPT               >          Old name  : DEPT
New owner  . . DSN8D10   > Type:   (U/R)      Old owner : DSN8D10
Partitions ==> 1         Action    : NONE     New DB    : DSN8D81A
Rows/Page    : 48.188                         New TS    : DSN8S81D
Commands :  NEXT CONSTRAINTS TBLOPTS  LONGNAMES            HASH
Line commands :
 I - Insert  U - Update  D - Delete  R - Repeat   LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index    RES - Reset update
 ? - Show all line commands
                                                                  Old Operation
Select Column Name        Col No Col Type Length  Scale Null D Col No Type
       *                       * *             *      * *    *      * *
------ ----------------- ------ -------- ------ ------ ---- - ------ ---------


       DEPTNO                  1 CHAR         3      0 N    N      1
I      DEPTNAME                2 VARCHAR     36      0 N    N      2
       MGRNO                   3 CHAR         6      0 Y    Y      3
       ADMRDEPT                4 CHAR         3      0 N    N      4
       LOCATION                5 CHAR        16      0 Y    Y      5
******************************* END OF DB2 DATA *******************************
```

Figure 249. ***ALTER Table (ADB27C)*** *panel- Inserting a column*

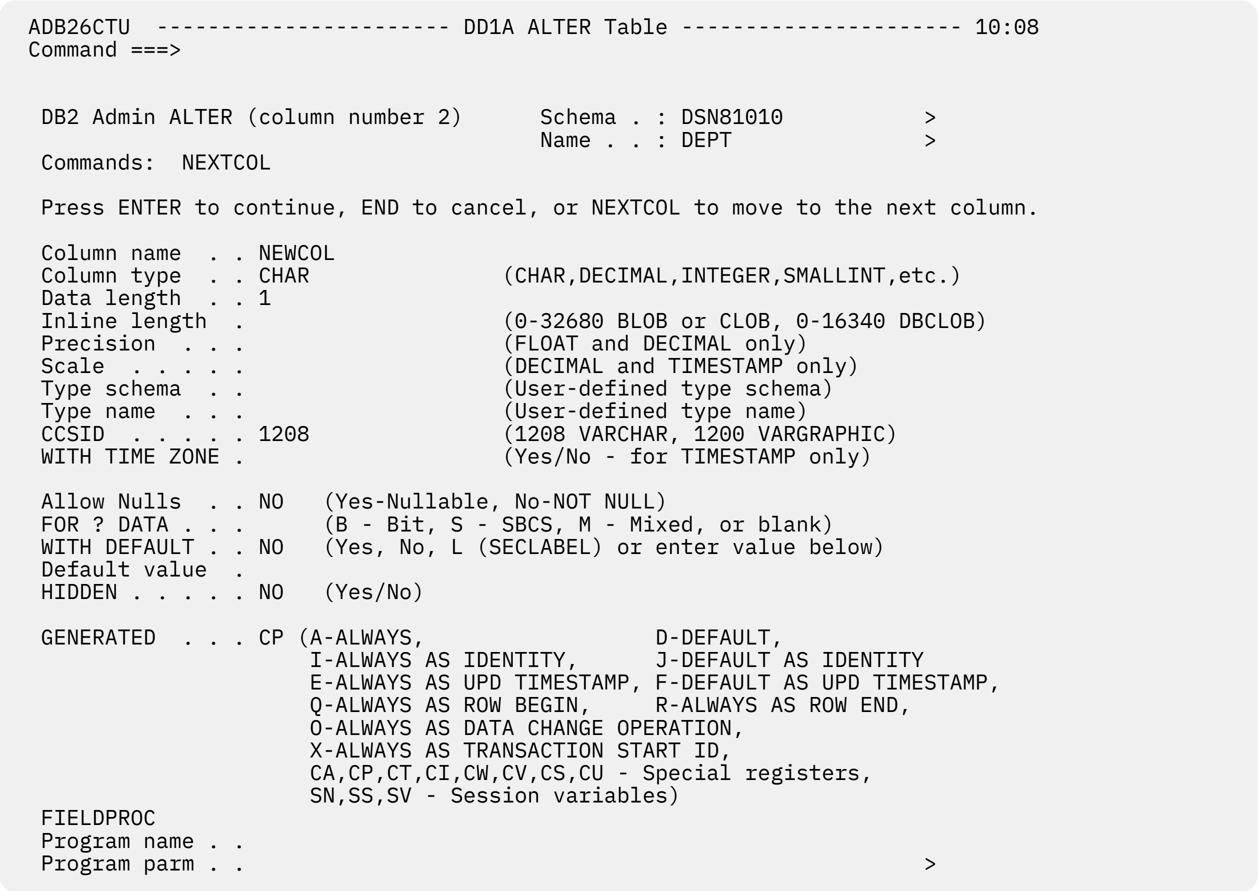
5. For the line that contains **INSERT** in the **Operation Type** column, complete the fields that contain **?** as shown to define the new column, and press Enter:



```
ADB27C in ------------------- DD1A ALTER Table --------------- Row 1 to 6 of 6
Command ===>                                                  Scroll ===> PAGE

New schema . . DSN8A10   >                    Old schema: DSN8A10
New name . . . DEPT               >          Old name  : DEPT
Partitions . : 0                              New DB  . . DSN8DA1A
Rows per page: 53                             New TS  . . DSN8SA1D

Commands:  CONTINUE  CONSTRAINTS TBLOPTS  LONGNAMES            HASH
Line commands:
 I - Insert  U - Update  D - Delete  R - Repeat  LAB - Label  COM - Comment
 M - Move    A - After   B - Before  X - Index   RES - Reset update
 ? - Show all line commands
                                                                 Old Operation
Sel Column Name        Col No Col Type      Length  Scale N D Col No Type
    *                       * *                  *      * * *      * *
--- -----------------> ------ -------- ----------- ------ - - ------ ---------

    DEPTNO                  1 CHAR               3      0 N N      1
*   DEPTNAME                2 VARCHAR           36      0 N N      2
    BUILDING                3 CHAR               0      0 ? ?      0 INSERT
    MGRNO                   4 CHAR               6      0 Y Y      3
    ADMRDEPT                5 CHAR               3      0 N N      4
    LOCATION                6 CHAR              16      0 Y Y      5
******************************* END OF DB2 DATA *******************************
```

Figure 250. ***ALTER Table (ADB27C)*** *panel - Specifying attributes for the inserted column*

6. If you want to specify additional attributes for the new column, use the U line command, and on the **ALTER Table (ADB26CTU)** panel, specify the updated values, and press Enter:



```
ADB26CTU  ----------------------- DD1A ALTER Table ---------------------- 10:08
Command ===>


 DB2 Admin ALTER (column number 2)      Schema . : DSN81010           >
                                        Name . . : DEPT               >
 Commands:  NEXTCOL

 Press ENTER to continue, END to cancel, or NEXTCOL to move to the next column.

 Column name  . . NEWCOL
 Column type  . . CHAR               (CHAR,DECIMAL,INTEGER,SMALLINT,etc.)
 Data length  . . 1
 Inline length  .                    (0-32680 BLOB or CLOB, 0-16340 DBCLOB)
 Precision  . . .                    (FLOAT and DECIMAL only)
 Scale  . . . . .                    (DECIMAL and TIMESTAMP only)
 Type schema  . .                    (User-defined type schema)
 Type name  . . .                    (User-defined type name)
 CCSID  . . . . . 1208               (1208 VARCHAR, 1200 VARGRAPHIC)
 WITH TIME ZONE .                    (Yes/No - for TIMESTAMP only)

 Allow Nulls  . . NO   (Yes-Nullable, No-NOT NULL)
 FOR ? DATA . . .      (B - Bit, S - SBCS, M - Mixed, or blank)
 WITH DEFAULT . . NO   (Yes, No, L (SECLABEL) or enter value below)
 Default value  .
 HIDDEN . . . . . NO   (Yes/No)

 GENERATED  . . . CP (A-ALWAYS,                 D-DEFAULT,
                      I-ALWAYS AS IDENTITY,      J-DEFAULT AS IDENTITY
                      E-ALWAYS AS UPD TIMESTAMP, F-DEFAULT AS UPD TIMESTAMP,
                      Q-ALWAYS AS ROW BEGIN,     R-ALWAYS AS ROW END,
                      O-ALWAYS AS DATA CHANGE OPERATION,
                      X-ALWAYS AS TRANSACTION START ID,
                      CA,CP,CT,CI,CW,CV,CS,CU - Special registers,
                      SN,SS,SV - Session variables)
 FIELDPROC
 Program name . .
 Program parm . .                                                     >
```

*Figure 251. **ALTER Table (ADB26CTU)** panel*

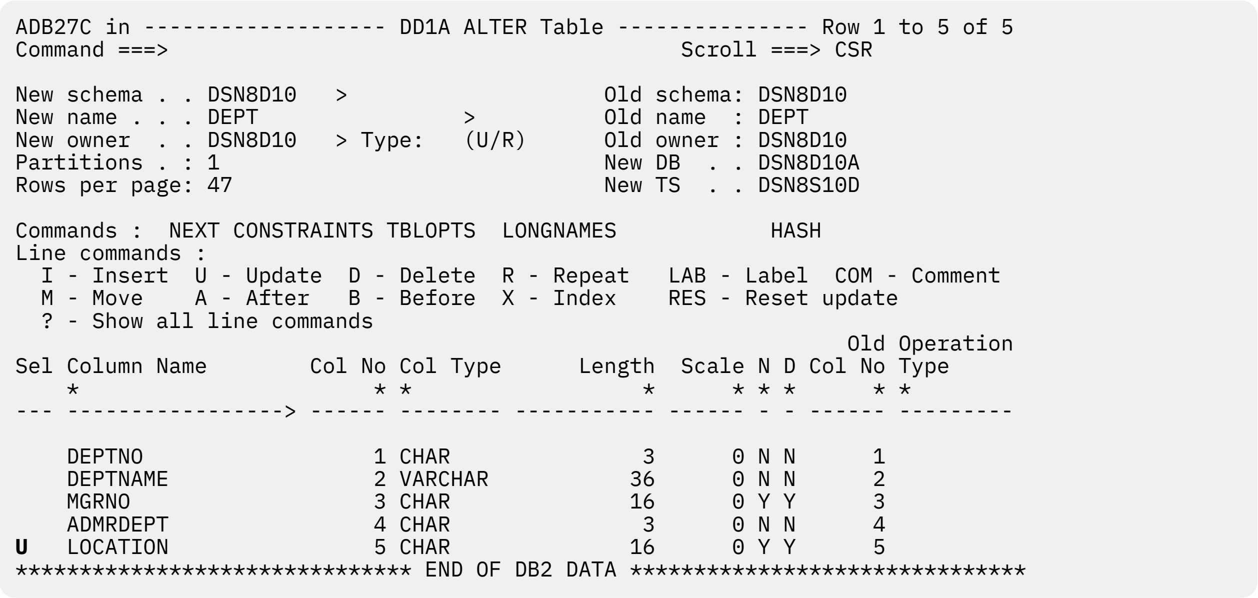
7. On the **ALTER Table (ADB27C)** panel, issue the NEXT command until the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed.
8. Choose options for building the WSL or batch job to implement the change, and press Enter to generate the job.

Updating a column

Procedure

To update a column:

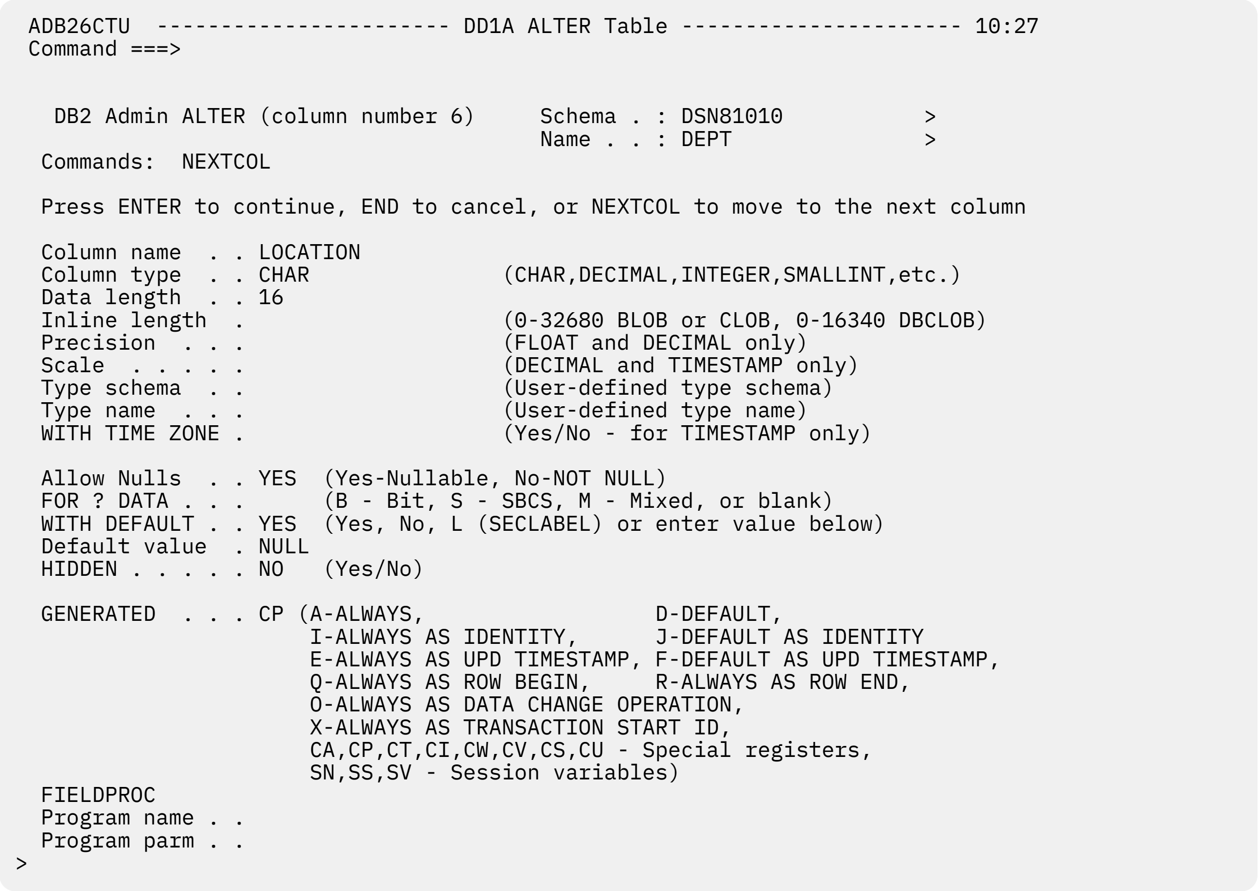
1. On the **DB2 Administration Menu (ADB2)** panel, select option 1.
2. On the **System Catalog (ADB21)** panel, select option T.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT line command against the table to be changed.
4. On the **ALTER Table (ADB27C)** panel, type over the fields for any column that you want to update.
5. If you want to update column attributes that are not displayed on the **ALTER Table (ADB27C)** panel, issue the U line command against the column:



```
ADB27C in ------------------- DD1A ALTER Table --------------- Row 1 to 5 of 5
Command ===>                                        Scroll ===> CSR

New schema . . DSN8D10   >                    Old schema: DSN8D10
New name . . . DEPT               >           Old name  : DEPT
New owner  . . DSN8D10   > Type:   (U/R)      Old owner : DSN8D10
Partitions . : 1                              New DB  . . DSN8D10A
Rows per page: 47                             New TS  . . DSN8S10D

Commands :  NEXT CONSTRAINTS TBLOPTS  LONGNAMES           HASH
Line commands :
  I - Insert  U - Update  D - Delete  R - Repeat   LAB - Label  COM - Comment
  M - Move    A - After   B - Before  X - Index    RES - Reset update
  ? - Show all line commands
                                                                Old Operation
Sel Column Name        Col No Col Type      Length  Scale N D Col No Type
    *                       * *                  *      * * *      * *
--- -----------------> ------ -------- ----------- ------ - - ------ ---------

    DEPTNO                  1 CHAR               3      0 N N      1
    DEPTNAME                2 VARCHAR           36      0 N N      2
    MGRNO                   3 CHAR              16      0 Y Y      3
    ADMRDEPT                4 CHAR               3      0 N N      4
U   LOCATION                5 CHAR              16      0 Y Y      5
******************************* END OF DB2 DATA *******************************
```

Figure 252. ***ALTER Table (ADB27C)*** *panel - Updating a column*

6. On the **ALTER Table (ADB26CTU)** panel, specify updated values for any attributes, and press Enter:



```
 ADB26CTU  ----------------------- DD1A ALTER Table ---------------------- 10:27
 Command ===>


   DB2 Admin ALTER (column number 6)     Schema . : DSN81010           >
                                         Name . . : DEPT               >
  Commands:  NEXTCOL

  Press ENTER to continue, END to cancel, or NEXTCOL to move to the next column

  Column name  . . LOCATION
  Column type  . . CHAR               (CHAR,DECIMAL,INTEGER,SMALLINT,etc.)
  Data length  . . 16
  Inline length  .                    (0-32680 BLOB or CLOB, 0-16340 DBCLOB)
  Precision  . . .                    (FLOAT and DECIMAL only)
  Scale  . . . . .                    (DECIMAL and TIMESTAMP only)
  Type schema  . .                    (User-defined type schema)
  Type name  . . .                    (User-defined type name)
  WITH TIME ZONE .                    (Yes/No - for TIMESTAMP only)

  Allow Nulls  . . YES  (Yes-Nullable, No-NOT NULL)
  FOR ? DATA . . .      (B - Bit, S - SBCS, M - Mixed, or blank)
  WITH DEFAULT . . YES  (Yes, No, L (SECLABEL) or enter value below)
  Default value  . NULL
  HIDDEN . . . . . NO   (Yes/No)

  GENERATED  . . . CP (A-ALWAYS,                 D-DEFAULT,
                      I-ALWAYS AS IDENTITY,      J-DEFAULT AS IDENTITY
                      E-ALWAYS AS UPD TIMESTAMP, F-DEFAULT AS UPD TIMESTAMP,
                      Q-ALWAYS AS ROW BEGIN,     R-ALWAYS AS ROW END,
                      O-ALWAYS AS DATA CHANGE OPERATION,
                      X-ALWAYS AS TRANSACTION START ID,
                      CA,CP,CT,CI,CW,CV,CS,CU - Special registers,
                      SN,SS,SV - Session variables)
  FIELDPROC
  Program name . .
  Program parm . .
>
```

Figure 253. ***ALTER Table (ADB26CTU)*** *panel*

If you altered a primary key column of a table, an additional primary command, ADDFK, is displayed on the **ALTER Table (ADB27C)** panel. ADDFK propagates the primary key column updates for the target table to all tables that are affected by the update. If you specify ADDFK, all affected tables are displayed on the **Alter Objects (ADB27CA)** panel and included in the subsequently generated JCL for the ALTER. Submit that JCL and skip the remaining steps.

7. On the **ALTER Table (ADB27C)** panel, issue the NEXT command until the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed.
8. Choose options for building the WSL or batch job to implement the change, and press Enter to generate the job.

Adding a unique key to a table

Procedure

To add a unique key to a table:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1.
2. On the **System Catalog (ADB21)** panel, select option T.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT line command against the table to which you want to add a unique key.
4. On the **ALTER Table (ADB27C)** panel, issue the CONSTRAINTS primary command.

   If any constraints exist on the table, the **Alter - Unique Constraints (ADBP7CN)** panel lists the primary key and unique key constraints.
5. If the **Alter - Unique Constraints (ADBP7CN)** panel is displayed, issue the ADD primary command.
6. On the **Create Primary or Unique Key (ADBP7CTP)** panel, specify the following options for the unique key, and issue the NEXT command:
   - In the **Constraint name** field, type a name for the constraint.
   - In the **Type** field, specify whether the key is a primary or a unique key.
   - For the columns in the table, use the *nn* line command to specify the relative position of the column in the key.
7. If the **ALTER Table (ADB27C)** panel is not displayed, press PF3 until that panel is displayed.
8. On the **ALTER Table (ADB27C)** panel, issue the NEXT command until the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed.
9. Choose options for building the WSL or batch job to implement the change, and press Enter to generate the job.

Changing a unique key

Procedure

To change a unique key:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1.
2. On the **System Catalog (ADB21)** panel, select option T.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT line command against the table for which you want to alter a unique key.
4. On the **ALTER Table (ADB27C)** panel, issue the CONSTRAINTS primary command.

   The **Alter - Unique Constraints (ADBP7CN)** panel lists the primary key and unique key constraints for the table. If this panel is not displayed, no constraints exist on the table.
5. Issue the **A** line command against the constraint that you want to alter.
6. On the resulting panel, either the **Alter Primary Key (ADBP7CTP)** panel or **Alter Unique Key (ADBP7CTP)** panel, specify the options that you want to change, and issue the NEXT command:
   - If you are changing a primary key, in the **Constraint name** field, type a new name.
   - Use the *nn* line command to change the relative position of the column in the key.
7. Press PF3 to return to the **ALTER Table (ADB27C)** panel.
8. On the **ALTER Table (ADB27C)** panel, issue the NEXT command until the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed.

9. Choose options for building the WSL or batch job to implement the change, and press Enter to generate the job.

Adding a column to a primary key

About this task

When you add a column to a primary key, the underlying index that enforces that constraint needs to be changed at the same time. (This index is called the *primary index*.) The complete change can be made using a single ALT command.

Procedure

To add a column to a primary key:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1.
2. On the **System Catalog (ADB21)** panel, select option T.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT line command against the table whose primary key you want to alter, and press Enter.
4. On the **ALTER Table (ADB27C)** panel, issue the CONSTRAINTS primary command, and press Enter.
5. On the **Alter - Unique Constraints (ADBP7CN)** panel, issue the A line command against the primary key that you want to alter, and press Enter.

   The **Alter Primary Key (ADBP7CTP)** panel displays the current primary key.

   In the following example, the key includes columns INT1 and INT2, in that order:

```
ADBP7CTP  --------------- DD1A Alter Primary Key ------------- Row 1 to 5 of 5

Table schema  . . . : TS6462
Table name  . . . . : PKTAB
Constraint name . . : INT1                > Type . . : PRIMARY

BUSINESS_TIME WITHOUT OVERLAPS . . . NO   (Yes/No)

Commands: NEXT
Line commands:  nn - Sequence  R - Remove column sequence

Select Column Name        Col Seq Col Type      Length  Scale Nulls Period
       *                        * *                  *      * *     *
------ ------------------ ------- -------- ----------- ------ ----- ------
       NAME                     0 CHAR              24      0 N
       INT1                     1 INTEGER            4      0 N
       INT2                     2 INTEGER            4      0 N
       INT3                     0 INTEGER            4      0 N
       INT4                     0 INTEGER            4      0 Y
****************************** END OF DB2 DATA *******************************
```

6. To add a column to the primary key, specify a sequence number in the **Select** column, and press Enter.

   For example, to add the INT3 column to the primary key, specify 3 next to INT3:

```
ADBP7CTP  --------------- DD1A Alter Primary Key ------------- Row 1 to 5 of 5

Table schema  . . . : TS6462
Table name  . . . . : PKTAB
Constraint name . . : INT1                > Type . . : PRIMARY

BUSINESS_TIME WITHOUT OVERLAPS . . . NO   (Yes/No)

Commands: NEXT
Line commands:  nn - Sequence  R - Remove column sequence

Select Column Name        Col Seq Col Type      Length  Scale Nulls Period
       *                        * *                  *      * *     *
------ ----------------- ------- -------- ----------- ------ ----- ------
       NAME                     0 CHAR              24      0 N
       INT1                     1 INTEGER            4      0 N
       INT2                     2 INTEGER            4      0 N
3      INT3                     0 INTEGER            4      0 N
       INT4                     0 INTEGER            4      0 Y
******************************* END OF DB2 DATA *******************************
```

After you press Enter, the **Seq** column is updated:

```
ADBP7CTP  --------------- DD1A Alter Primary Key ------------- Row 1 to 5 of 5

Table schema  . . . : TS6462
Table name  . . . . : PKTAB
Constraint name . . : INT1                > Type . . : PRIMARY

BUSINESS_TIME WITHOUT OVERLAPS . . . NO   (Yes/No)

Commands: NEXT
Line commands:  nn - Sequence  R - Remove column sequence

Select Column Name        Col Seq Col Type      Length  Scale Nulls Period
       *                        * *                  *      * *     *
------ ----------------- ------- -------- ----------- ------ ----- ------
       NAME                     0 CHAR              24      0 N
       INT1                     1 INTEGER            4      0 N
       INT2                     2 INTEGER            4      0 N
*      INT3                     3 INTEGER            4      0 N
       INT4                     0 INTEGER            4      0 Y
******************************* END OF DB2 DATA *******************************
```

You can also make other changes on this panel, such as changing the name of the constraint or removing columns from the sequence. You can make multiple changes to the table during this ALT session.

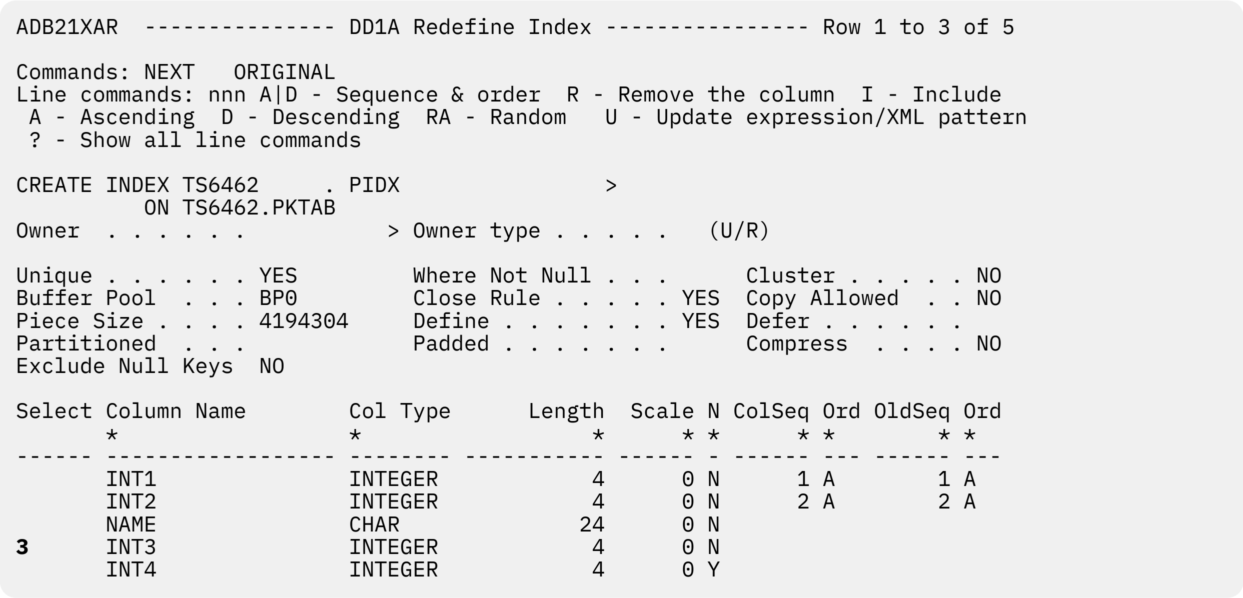
7. Issue the NEXT command, and press Enter.
8. On the **Alter - Unique Constraints (ADBP7CN)** panel, press PF3 to exit the panel.
9. On the **ALTER Table (ADB27C)** panel, issue the NEXT command and press Enter.

   The following warning message is displayed:

   ```
   Primary index required.
   ```

   This warning is a reminder that you must also alter the primary index to include the added column.
10. On the **ALT - Related Objects (ADBP7REL)** panel, specify the A line command next to the index to add it to the list of altered objects, and press Enter.
11. Press PF3 to exit the panel.
12. On the **Alter Objects (ADB27CA)** panel, specify the A line command next to the index, and press Enter.
13. On the **Redefine Index (ADB21XAR)** panel, specify a sequence number next to the column that you just added to the primary key, and press Enter.

    For example, specify 3 next to INT3:



```
ADB21XAR  --------------- DD1A Redefine Index ---------------- Row 1 to 3 of 5

Commands: NEXT   ORIGINAL
Line commands: nnn A|D - Sequence & order  R - Remove the column  I - Include
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern
 ? - Show all line commands

CREATE INDEX TS6462     . PIDX                >
          ON TS6462.PKTAB
Owner  . . . . . .           > Owner type . . . . .   (U/R)

Unique . . . . . . YES         Where Not Null . . .      Cluster . . . . . NO
Buffer Pool  . . . BP0         Close Rule . . . . . YES  Copy Allowed  . . NO
Piece Size . . . . 4194304     Define . . . . . . . YES  Defer . . . . . .
Partitioned  . . .             Padded . . . . . . .      Compress  . . . . NO
Exclude Null Keys  NO

Select Column Name        Col Type      Length  Scale N ColSeq Ord OldSeq Ord
       *                  *                  *      * *      * *        * *
------ ------------------ -------- ----------- ------ - ------ --- ------ ---
       INT1               INTEGER            4      0 N      1 A        1 A
       INT2               INTEGER            4      0 N      2 A        2 A
       NAME               CHAR              24      0 N
3      INT3               INTEGER            4      0 N
       INT4               INTEGER            4      0 Y
```

After you press Enter, the **ColSeq** column is updated.

14. Issue NEXT until the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed:
15. Choose options for building the WSL or batch job to implement the change, and press Enter to generate the job.

Renaming a table

Procedure

To rename a table:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1.
2. On the **System Catalog (ADB21)** panel, select option T.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the REN line command against the table that you want to rename.
4. On the **Rename Table (ADB21TR)** panel, in the **New name** field, specify the new name, and press Enter.
5. If the **Statement Execution Prompt (ADB2PSTM)** panel is displayed, enter the appropriate action to execute the RENAME statement.

Adding a partition to a table

Procedure

To add a partition to a table:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1.
2. On the **System Catalog (ADB21)** panel, select option T.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT line command against the partitioned table to which you want to add a partition.
4. On the **ALTER Table (ADB27C)** panel, issue the ALTPART primary command to add or alter a partition of a table-based partitioned table, and press Enter.

   If the ALTPART command is not listed, the table is not partitioned.
5. On the **Alter Partitioned Table (ADB27CPV)** panel, issue the ADD primary command to add a partition to the end of the table.
6. Enter the limit key value for the new partition.

7. Issue the NEXT command.
8. On the **ALTER Table (ADB27C)** panel, issue the NEXT command until the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed.
9. Choose options for building the WSL or batch job to implement the change, and press Enter to generate the job.

Adding a partition to a table in a partition-by-growth (PBG) table space

Procedure

To add a partition to a table in a PBG table space:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1.
2. On the **System Catalog (ADB21)** panel, select option T.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT line command against the partitioned table to which you want to add a partition.
4. On the **ALTER Table (ADB27C)** panel, issue the ADDPART primary command, and press Enter.

   The **Partitions** field is updated to reflect the change. Specifying a zero (ADDPART 0) resets the number of partitions to the original value.
5. Issue the NEXT command until the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed.
6. Choose options for building the WSL or batch job to implement the change, and press Enter to generate the job.

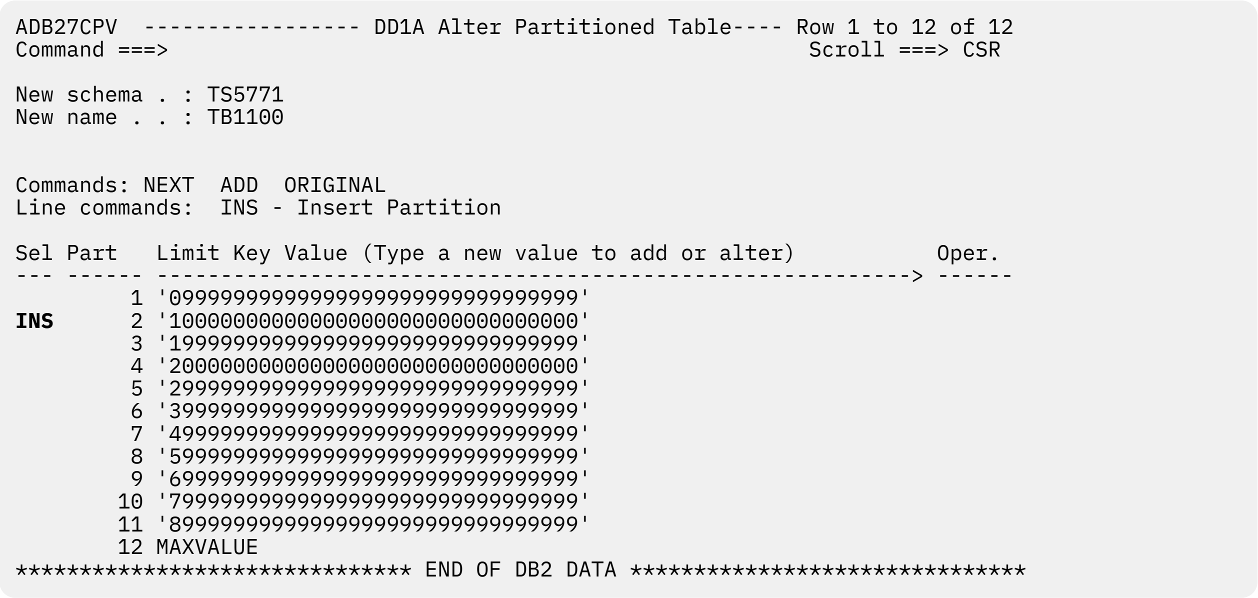
Inserting a partition into a table

Procedure

To insert a partition into a table:

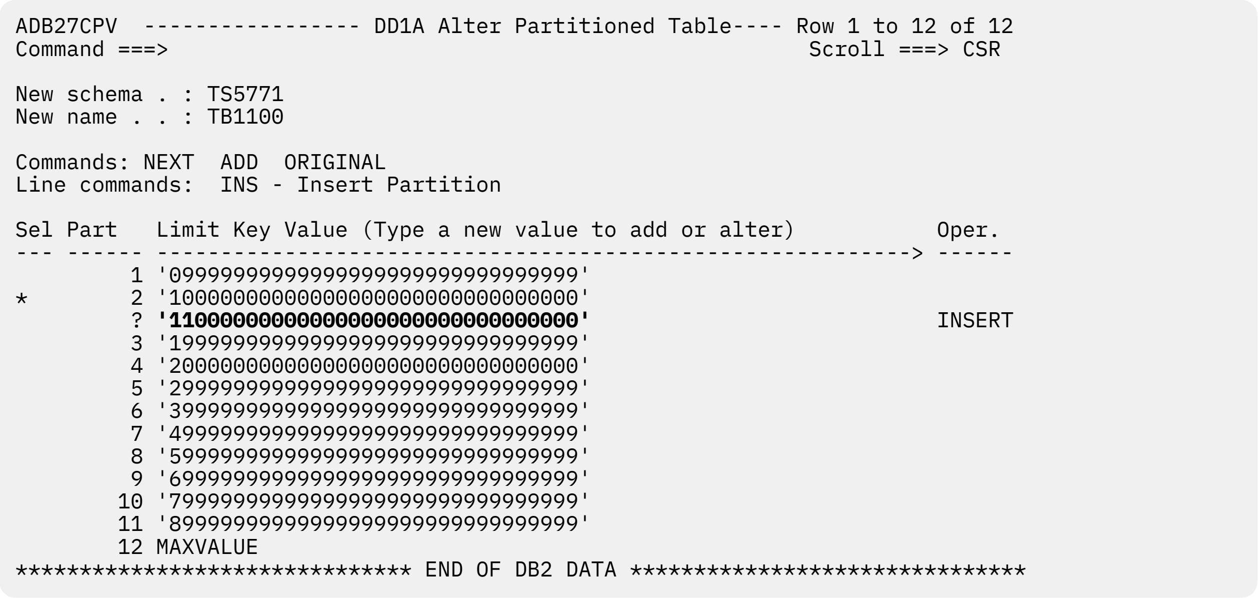
1. On the **DB2 Administration Menu (ADB2)** panel, select option 1.
2. On the **System Catalog (ADB21)** panel, select option T.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT line command against the partitioned table where you want to insert a partition.
4. On the **ALTER Table (ADB27C)** panel, specify the ALTPART primary command, and press Enter.

   If the ALTPART command is not listed, the table is not partitioned.
5. On the **Alter Partitioned Table (ADB27CPV)** panel, issue the INS line command against the partition after which you want to insert a new partition, as shown in the following example:



```
ADB27CPV  ----------------- DD1A Alter Partitioned Table---- Row 1 to 12 of 12
Command ===>                                                  Scroll ===> CSR

New schema . : TS5771
New name . . : TB1100


Commands: NEXT  ADD  ORIGINAL
Line commands:  INS - Insert Partition

Sel Part   Limit Key Value (Type a new value to add or alter)           Oper.
--- ------ -----------------------------------------------------------> ------
         1 '09999999999999999999999999999999'
INS      2 '10000000000000000000000000000000'
         3 '19999999999999999999999999999999'
         4 '20000000000000000000000000000000'
         5 '29999999999999999999999999999999'
         6 '39999999999999999999999999999999'
         7 '49999999999999999999999999999999'
         8 '59999999999999999999999999999999'
         9 '69999999999999999999999999999999'
        10 '79999999999999999999999999999999'
        11 '89999999999999999999999999999999'
        12 MAXVALUE
******************************* END OF DB2 DATA *******************************
```

6. Enter the limit key value for the new partition, as shown in the following example:



```
ADB27CPV  ----------------- DD1A Alter Partitioned Table---- Row 1 to 12 of 12
Command ===>                                                  Scroll ===> CSR

New schema . : TS5771
New name . . : TB1100

Commands: NEXT  ADD  ORIGINAL
Line commands:  INS - Insert Partition

Sel Part   Limit Key Value (Type a new value to add or alter)           Oper.
--- ------ -----------------------------------------------------------> ------
         1 '09999999999999999999999999999999'
*        2 '10000000000000000000000000000000'
         ? '11000000000000000000000000000000'                           INSERT
         3 '19999999999999999999999999999999'
         4 '20000000000000000000000000000000'
         5 '29999999999999999999999999999999'
         6 '39999999999999999999999999999999'
         7 '49999999999999999999999999999999'
         8 '59999999999999999999999999999999'
         9 '69999999999999999999999999999999'
        10 '79999999999999999999999999999999'
        11 '89999999999999999999999999999999'
        12 MAXVALUE
******************************* END OF DB2 DATA *******************************
```

Tip: If, after editing the limit key values, you decide that you want to reset all changes to the original values, use the RESET command.

7. Issue the NEXT command.
8. On the **ALTER Table (ADB27C)** panel, issue the NEXT command until the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed.
9. Choose options for building the WSL or batch job to implement the change, and press Enter to generate the job.

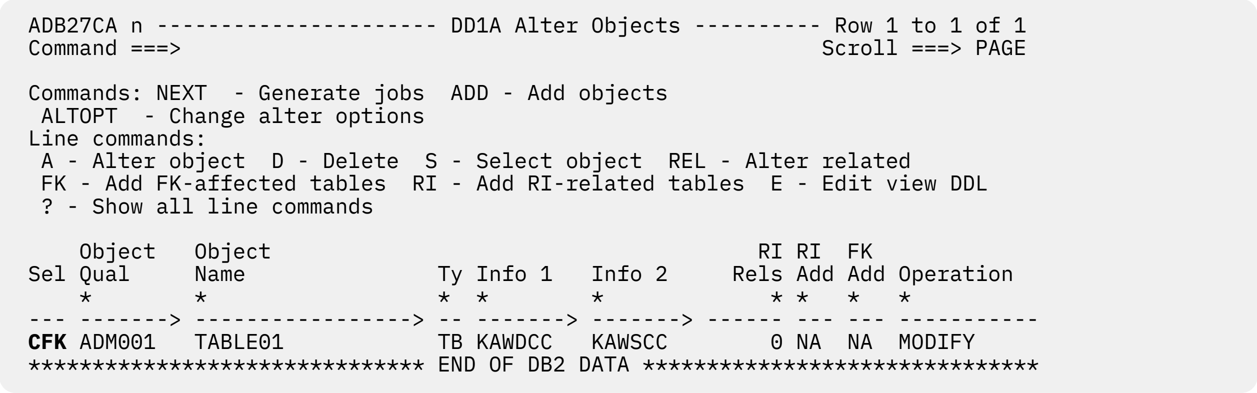
Adding a foreign key

Procedure

To add a foreign key:

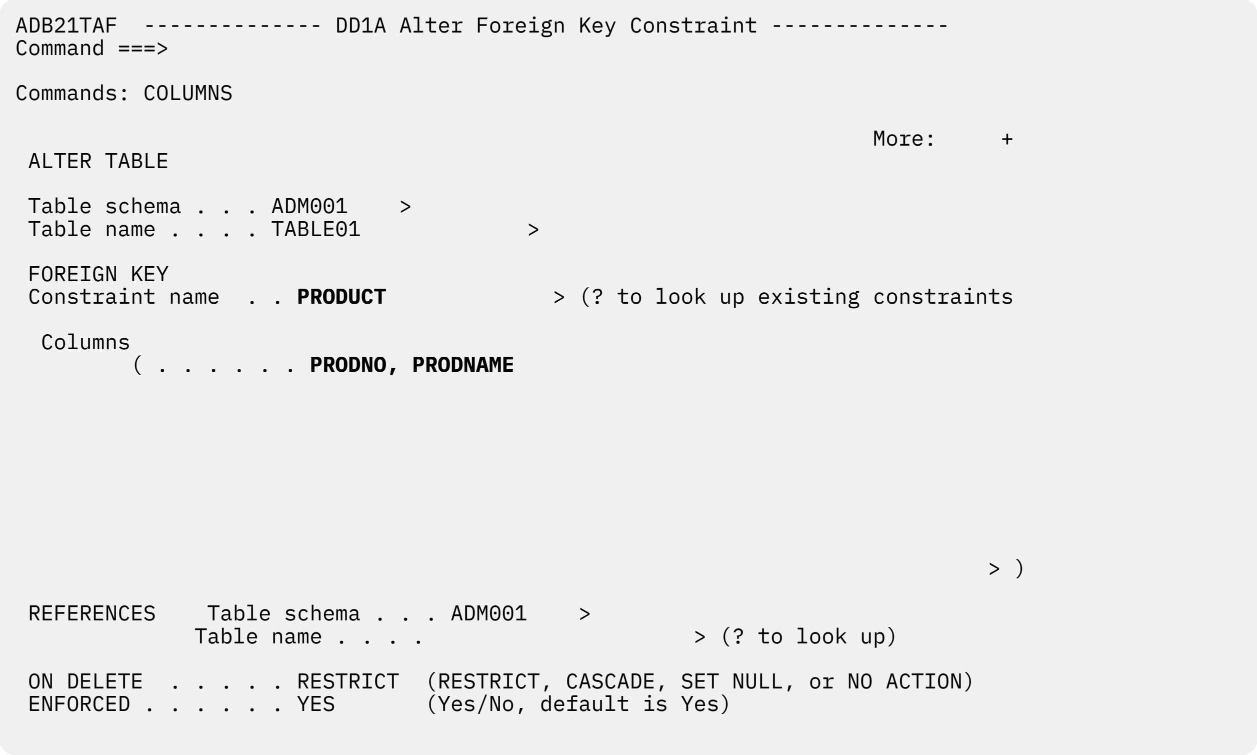
1. On the **DB2 Administration Menu (ADB2)** panel, select option 1.
2. On the **System Catalog (ADB21)** panel, select option T.

3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT line command against the table to be changed.
4. On the **ALTER Table (ADB27C)** panel, specify the NEXT primary command, and press Enter.
5. On the **Alter Objects (ADB27CA)** panel, specify the CFK (Create foreign key) line command next to the table where you want to add the foreign key.



```
ADB27CA n ---------------------- DD1A Alter Objects ---------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Commands: NEXT  - Generate jobs  ADD - Add objects
 ALTOPT  - Change alter options
Line commands:
 A - Alter object  D - Delete  S - Select object  REL - Alter related
 FK - Add FK-affected tables  RI - Add RI-related tables  E - Edit view DDL
 ? - Show all line commands

    Object   Object                                  RI RI  FK
Sel Qual     Name               Ty Info 1   Info 2     Rels Add Add Operation
    *        *                  *  *        *             * *   *   *
--- -------> -----------------> -- -------> -------> ------ --- --- -----------
CFK ADM001   TABLE01            TB KAWDCC   KAWSCC        0 NA  NA  MODIFY
******************************* END OF DB2 DATA *******************************
```

6. On the **Alter Foreign Key Constraint (ADB21TAF)** panel, specify the name of the constraint and the columns, and press Enter:



```
ADB21TAF  -------------- DD1A Alter Foreign Key Constraint --------------
Command ===>

Commands: COLUMNS

                                                                   More:     +
 ALTER TABLE

 Table schema . . . ADM001    >
 Table name . . . . TABLE01             >

 FOREIGN KEY
 Constraint name  . . PRODUCT             > (? to look up existing constraints

  Columns
         ( . . . . . . PRODNO, PRODNAME








                                                                            > )

 REFERENCES    Table schema . . . ADM001    >
              Table name . . . .                     > (? to look up)

 ON DELETE  . . . . . RESTRICT  (RESTRICT, CASCADE, SET NULL, or NO ACTION)
 ENFORCED . . . . . . YES       (Yes/No, default is Yes)
```

The ALTER TABLE statement is generated.

7. If the **Statement Execution Prompt (ADB2PSTM)** panel is displayed, confirm that you want to execute the statement.

Results

The ALTER statement is run, and the foreign key is added.

Changing the related objects for a table

When altering a table in Db2 Admin Tool, you can easily view related objects, such as table spaces, databases, indexes, views, and foreign keys. You can alter those objects as well.

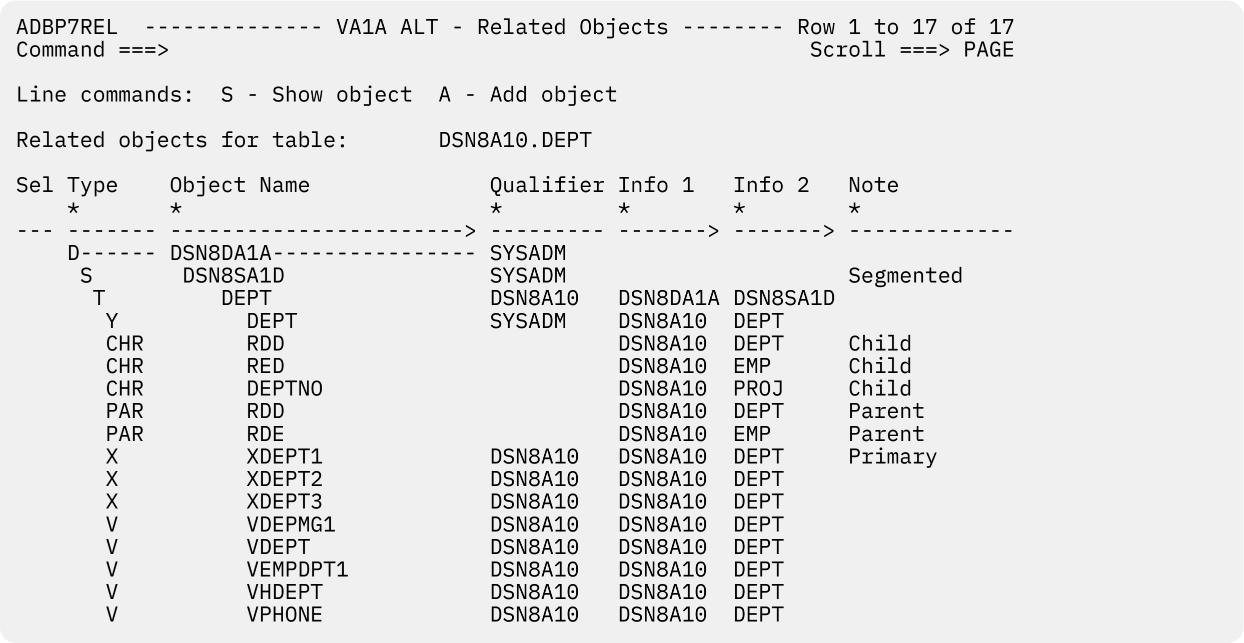
Procedure

To change the related objects for a table:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ALT command against the table whose related objects you want to change.
4. On the **ALTER Table (ADB27C)** panel, issue the NEXT command.
5. On the **Alter Objects (ADB27CA)** panel, issue the REL line command against the table whose related objects you want to change.

   The **ALT - Related Objects (ADBP7REL)** panel displays a list of related objects.
6. Optional: If the type of object that you want to change is not listed (for example, no indexes are listed), take the following actions to add those objects to this list:
   a) Exit back to the **Alter Objects (ADB27CA)** panel.
   b) Issue the ALTOPT command.
   c) On the **ALTER Options Menu (ADBP7OP)** panel, specify option 2 (REL options).
   d) On the **ALTER Choose Related Objects (ADBP7OBJ)** panel, specify YES for the object type that you want to include
   e) Issue the NEXT command.
   f) Exit back to the **Alter Objects (ADB27CA)** panel.
   g) Issue the REL line command against the table again. The list should now include the objects that you requested.
7. On the **ALT - Related Objects (ADBP7REL)** panel, issue the A line command against the object that you want to change.

   The A line command adds the object to the list of objects on the **Alter Objects (ADB27CA)** panel



```
ADBP7REL  -------------- VA1A ALT - Related Objects -------- Row 1 to 17 of 17
Command ===>                                                  Scroll ===> PAGE

Line commands:  S - Show object  A - Add object

Related objects for table:       DSN8A10.DEPT

Sel Type    Object Name              Qualifier Info 1   Info 2   Note
    *       *                        *         *        *        *
--- ------- -----------------------> --------- -------> -------> -------------
    D------ DSN8DA1A---------------- SYSADM
     S       DSN8SA1D                SYSADM                        Segmented
      T         DEPT                 DSN8A10   DSN8DA1A DSN8SA1D
       Y          DEPT               SYSADM    DSN8A10  DEPT
       CHR        RDD                          DSN8A10  DEPT     Child
       CHR        RED                          DSN8A10  EMP      Child
       CHR        DEPTNO                       DSN8A10  PROJ     Child
       PAR        RDD                          DSN8A10  DEPT     Parent
       PAR        RDE                          DSN8A10  EMP      Parent
       X          XDEPT1             DSN8A10   DSN8A10  DEPT     Primary
       X          XDEPT2             DSN8A10   DSN8A10  DEPT
       X          XDEPT3             DSN8A10   DSN8A10  DEPT
       V          VDEPMG1            DSN8A10   DSN8A10  DEPT
       V          VDEPT              DSN8A10   DSN8A10  DEPT
       V          VEMPDPT1           DSN8A10   DSN8A10  DEPT
       V          VHDEPT             DSN8A10   DSN8A10  DEPT
       V          VPHONE             DSN8A10   DSN8A10  DEPT
```

*Figure 254. **ALT - Related Objects (ADBP7REL)** panel*

8. Exit back to the **Alter Objects (ADB27CA)** panel.
9. On the **Alter Objects (ADB27CA)** panel, use the A line command to alter any object.
10. Specify the changes to the object.
11. Exit back to the **Alter Objects (ADB27CA)** panel.
12. Generate an ALT job

Related tasks

“Altering databases” on page 450
You can change certain database attributes, including the name of the database. Depending on the change that you want to make, use either the AL or ALT line command.

“Altering table spaces” on page 452
You can make changes to table spaces or partitions, such as changing the name or lock size or converting a table space to be a partitioned. Depending on the change that you want to make, use either the AL or ALT line command.

“Altering tables” on page 460
You can make changes to tables, such as adding columns, changing the name, or dropping constraints. Depending on the change that you want to make, use either the AL or ALT line command.

“Altering indexes” on page 490
You can rename an index, change index attributes, and change the columns that are indexed. Depending on the change that you want to make, use either the AL or ALT line command.

Altering a partition in an archive table

In Db2 Admin Tool, you can make changes to the partition configuration of an archive table by altering its table space. Db2 Admin Tool restricts altering an archive table directly, because the archive table should match its base table, the archive-enabled table.

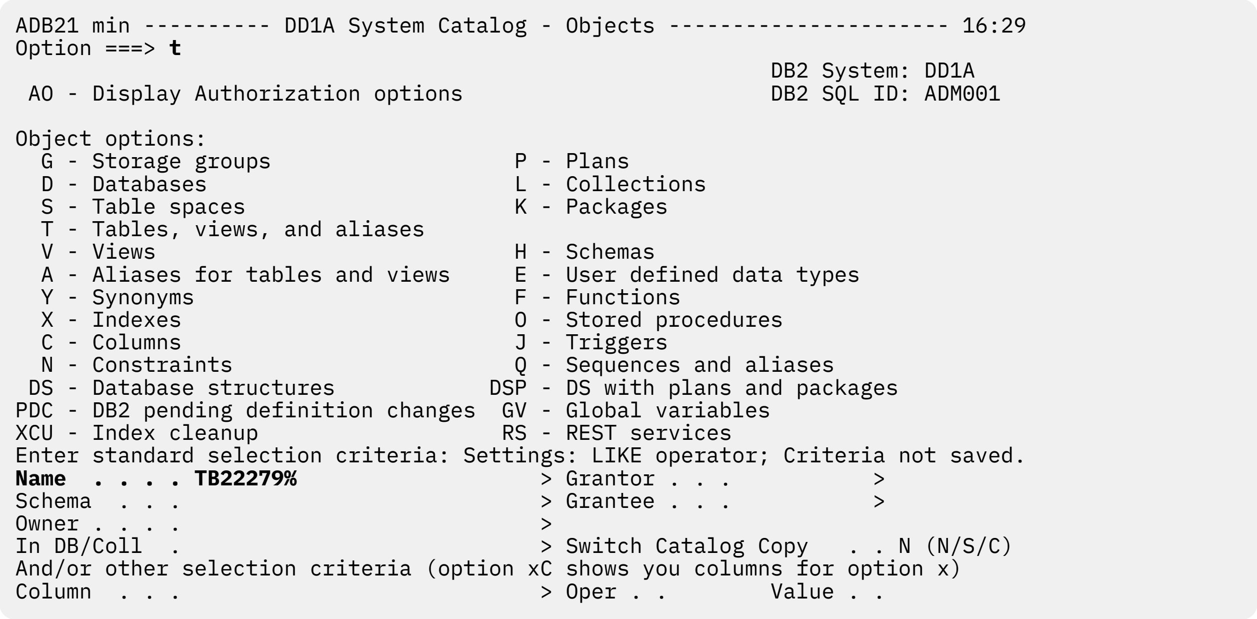
You might need to alter the partitions in an archive table when making changes to the base table, the archive-enabled table. With APAR PH42871, any partition changes that are made for an archive-enabled table are not automatically applied to its associated archive table to ensure the preservation of historical data. Therefore, if you need to apply these partition changes to the archive table, use the ALT line command on the table space as shown in the following procedure.

About this task

The following procedure explains how to remove partitions from the archive table. The sample scenario in these steps shows an archive table that has 5 partitions with the limit key values ‘A’, ‘B’, ‘C’, ‘D’, and ‘E’. The scenario removes the partitions with limit key values of ‘C’ and ‘D’.

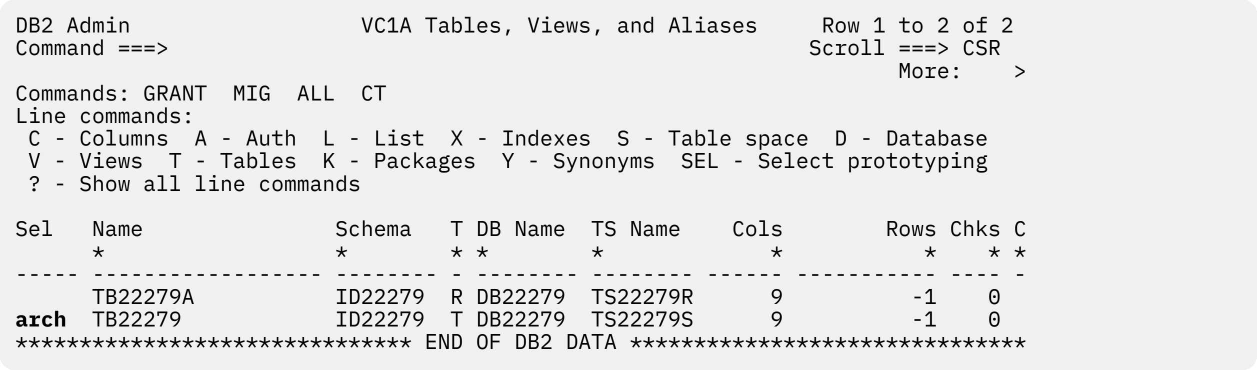
Procedure

1. Locate the archive-enabled table:
   a) On **DB2 Administration Menu (ADB2)** panel, select option 1, and press Enter.
   b) On **System Catalog (ADB21)** panel, specify option T and the name of the archive-enabled table in the **Name** field, and press Enter.



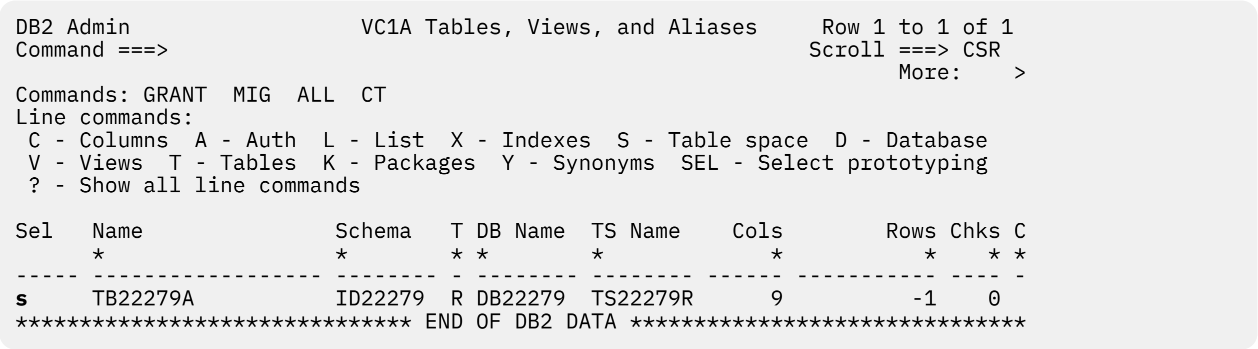
```
ADB21 min ---------- DD1A System Catalog - Objects --------------------- 16:29
Option ===> t
                                                           DB2 System: DD1A
 AO - Display Authorization options                        DB2 SQL ID: ADM001

Object options:
  G - Storage groups                    P - Plans
  D - Databases                         L - Collections
  S - Table spaces                      K - Packages
  T - Tables, views, and aliases
  V - Views                             H - Schemas
  A - Aliases for tables and views      E - User defined data types
  Y - Synonyms                          F - Functions
  X - Indexes                           O - Stored procedures
  C - Columns                           J - Triggers
  N - Constraints                       Q - Sequences and aliases
 DS - Database structures             DSP - DS with plans and packages
PDC - DB2 pending definition changes   GV - Global variables
XCU - Index cleanup                    RS - REST services
Enter standard selection criteria: Settings: LIKE operator; Criteria not saved.
Name  . . . . TB22279%                > Grantor . . .           >
Schema  . . .                         > Grantee . . .           >
Owner . . . .                         >
In DB/Coll  .                         > Switch Catalog Copy   . . N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .                         > Oper . .        Value . .
```

2. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the ARCH (show archive table information) line command against the archive-enabled table.



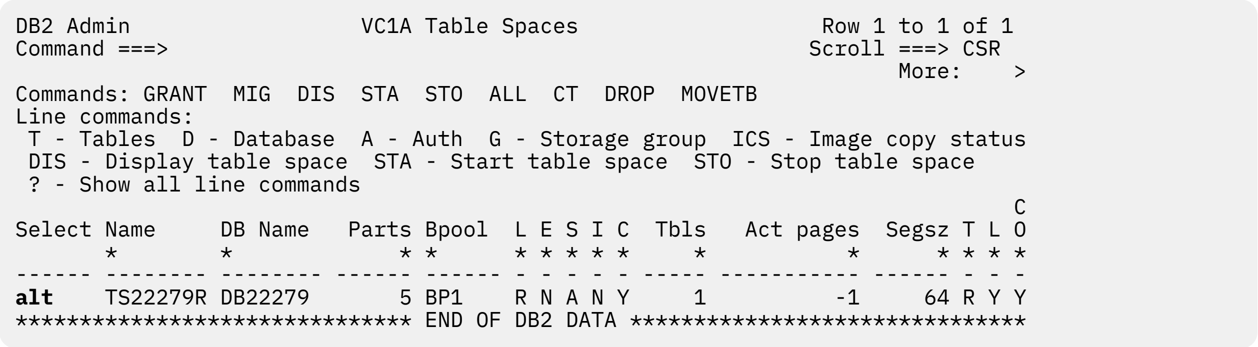
```
DB2 Admin                  VC1A Tables, Views, and Aliases     Row 1 to 2 of 2
Command ===>                                                  Scroll ===> CSR
                                                                     More:    >
Commands: GRANT  MIG  ALL  CT
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  K - Packages  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C
      *                  *        * *        *             *           *    * *
----- ------------------ -------- - -------- -------- ------ ----------- ---- -
      TB22279A           ID22279  R DB22279  TS22279R      9          -1    0
arch  TB22279            ID22279  T DB22279  TS22279S      9          -1    0
****************************** END OF DB2 DATA ******************************
```

3. On the updated **Tables, Views, and Aliases (ADB21T)** panel, issue the S (table space) line command against the archive table.



```
DB2 Admin                  VC1A Tables, Views, and Aliases     Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR
                                                                     More:    >
Commands: GRANT  MIG  ALL  CT
Line commands:
 C - Columns  A - Auth  L - List  X - Indexes  S - Table space  D - Database
 V - Views  T - Tables  K - Packages  Y - Synonyms  SEL - Select prototyping
 ? - Show all line commands

Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C
      *                  *        * *        *             *           *    * *
----- ------------------ -------- - -------- -------- ------ ----------- ---- -
s     TB22279A           ID22279  R DB22279  TS22279R      9          -1    0
******************************* END OF DB2 DATA *******************************
```

4. On the **Table Spaces (ADB21S)** panel, enter the ALT line command against the table space.

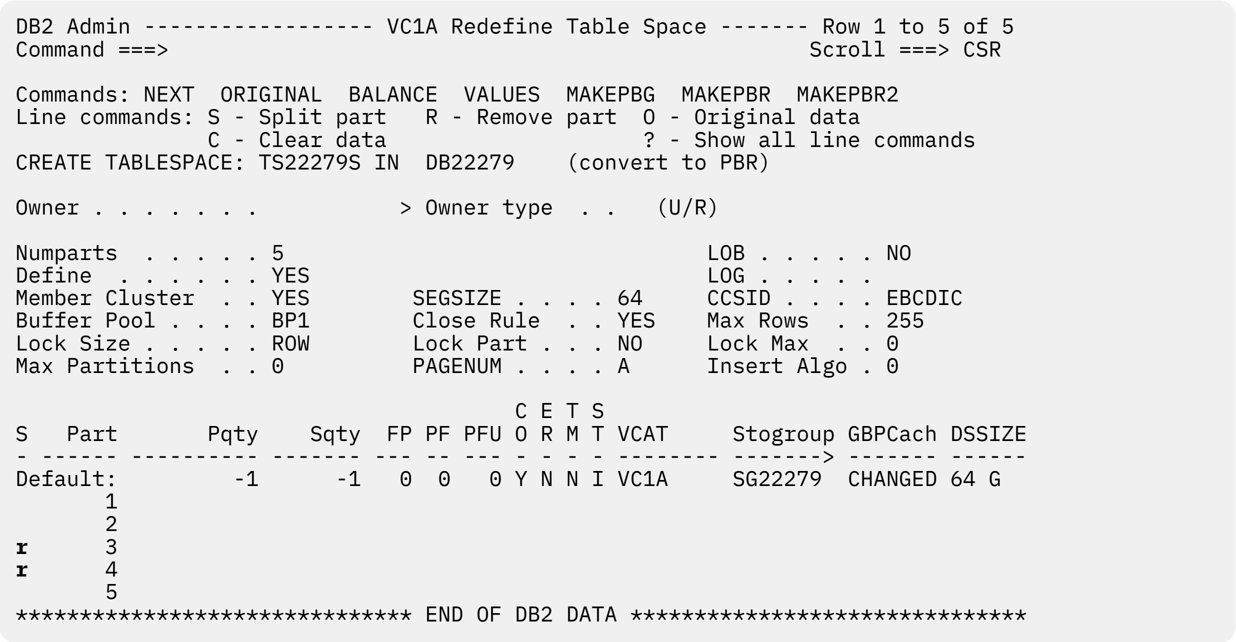


```
DB2 Admin                  VC1A Table Spaces                   Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR
                                                                     More:    >
Commands: GRANT  MIG  DIS  STA  STO  ALL  CT  DROP  MOVETB
Line commands:
 T - Tables  D - Database  A - Auth  G - Storage group  ICS - Image copy status
 DIS - Display table space  STA - Start table space  STO - Stop table space
 ? - Show all line commands
                                                                              C
Select Name     DB Name   Parts Bpool  L E S I C  Tbls   Act pages  Segsz T L O
       *        *             * *      * * * * *     *           *      * * * *
------ -------- -------- ------ ------ - - - - - ----- ---------- ------ - - -
alt    TS22279R DB22279       5 BP1    R N A N Y     1          -1     64 R Y Y
******************************* END OF DB2 DATA *******************************
```

For this scenario, specify No on the Change Management prompt

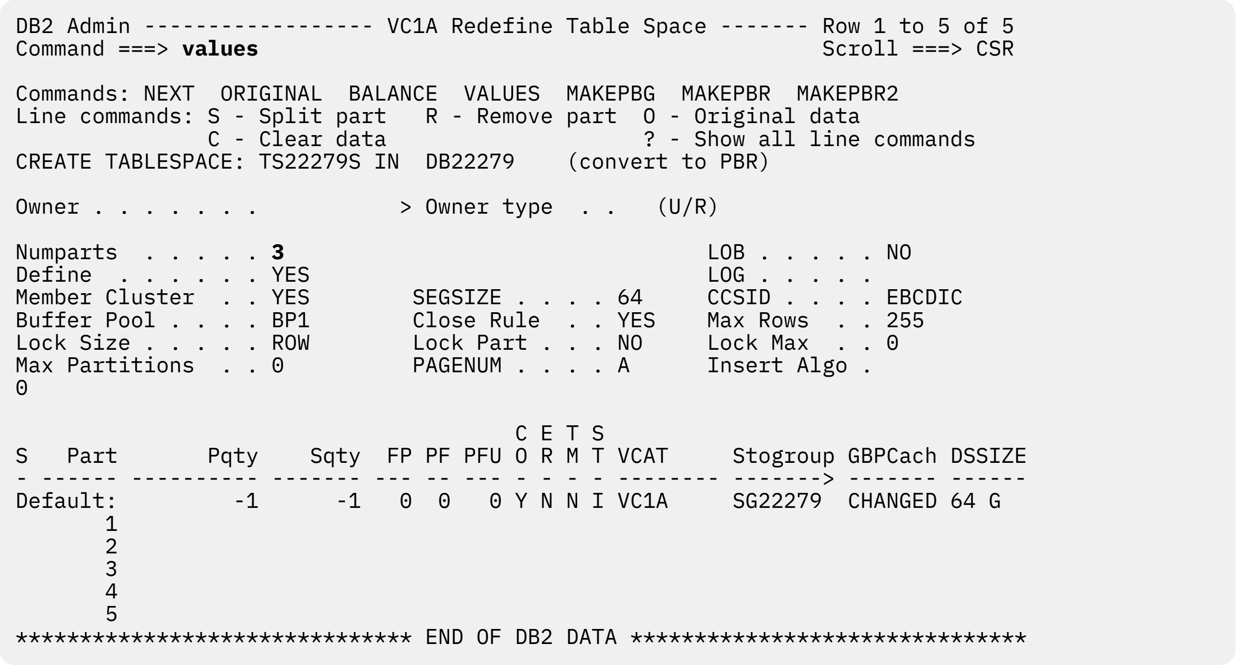
5. On the **Redefine Table Space (ADB21SAR)** panel, use one of the following two ways to remove a partition:

   - Use the R line command. The following example specifies that partitions 3 and 4 are to be removed:



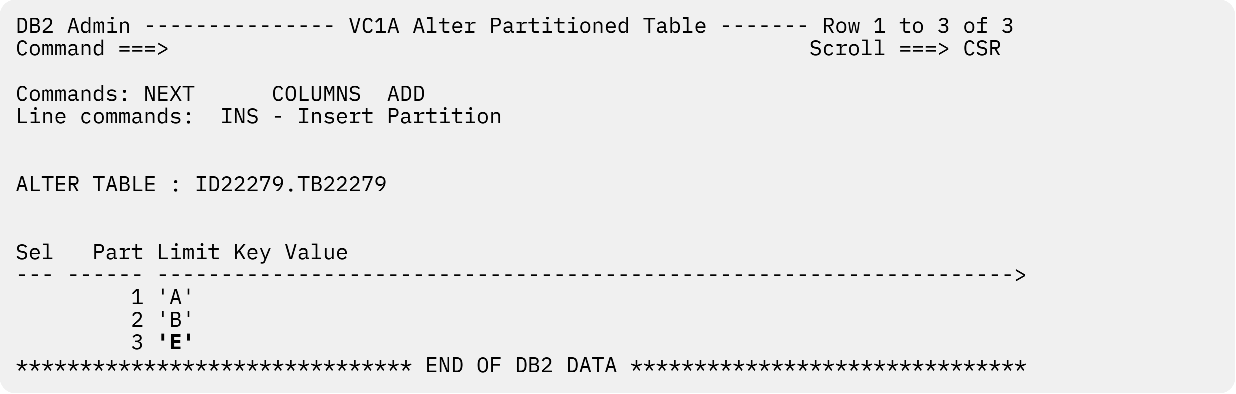
```
DB2 Admin ------------------ VC1A Redefine Table Space ------- Row 1 to 5 of 5
Command ===>                                                  Scroll ===> CSR

Commands: NEXT  ORIGINAL  BALANCE  VALUES  MAKEPBG  MAKEPBR  MAKEPBR2
Line commands: S - Split part   R - Remove part  O - Original data
               C - Clear data                    ? - Show all line commands
CREATE TABLESPACE: TS22279S IN  DB22279    (convert to PBR)

Owner . . . . . . .           > Owner type  . .   (U/R)

Numparts  . . . . . 5                                 LOB . . . . . NO
Define  . . . . . . YES                               LOG . . . . .
Member Cluster  . . YES        SEGSIZE . . . . 64     CCSID . . . . EBCDIC
Buffer Pool . . . . BP1        Close Rule  . . YES    Max Rows  . . 255
Lock Size . . . . . ROW        Lock Part . . . NO     Lock Max  . . 0
Max Partitions  . . 0          PAGENUM . . . . A      Insert Algo . 0

                                       C E T S
S   Part       Pqty    Sqty  FP PF PFU O R M T VCAT     Stogroup GBPCach DSSIZE
- ------ ---------- ------- --- -- --- - - - - -------- -------> ------- ------
Default:         -1      -1   0  0   0 Y N N I VC1A     SG22279  CHANGED 64 G
       1
       2
r      3
r      4
       5
******************************* END OF DB2 DATA *******************************
```

   - Change the value in the **Numparts** field and then use the VALUES command. In the following example, the **Numparts** value is changed from 5 to 3:



```
DB2 Admin ------------------ VC1A Redefine Table Space ------- Row 1 to 5 of 5
Command ===> values                                            Scroll ===> CSR

Commands: NEXT  ORIGINAL  BALANCE  VALUES  MAKEPBG  MAKEPBR  MAKEPBR2
Line commands: S - Split part   R - Remove part  O - Original data
               C - Clear data                    ? - Show all line commands
CREATE TABLESPACE: TS22279S IN  DB22279    (convert to PBR)

Owner . . . . . . .           > Owner type  . .   (U/R)

Numparts  . . . . . 3                                 LOB . . . . . NO
Define  . . . . . . YES                               LOG . . . . .
Member Cluster  . . YES        SEGSIZE . . . . 64     CCSID . . . . EBCDIC
Buffer Pool . . . . BP1        Close Rule  . . YES    Max Rows  . . 255
Lock Size . . . . . ROW        Lock Part . . . NO     Lock Max  . . 0
Max Partitions  . . 0          PAGENUM . . . . A      Insert Algo .
0

                                       C E T S
S   Part       Pqty    Sqty  FP PF PFU O R M T VCAT     Stogroup GBPCach DSSIZE
- ------ ---------- ------- --- -- --- - - - - -------- -------> ------- ------
Default:         -1      -1   0  0   0 Y N N I VC1A     SG22279  CHANGED 64 G
       1
       2
       3
       4
       5
******************************* END OF DB2 DATA *******************************
```

On the **Alter Partitioned Table (ADB21TAV)** panel, change the limit key value of the last partition to 'E' .



```
DB2 Admin --------------- VC1A Alter Partitioned Table ------- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Commands: NEXT      COLUMNS  ADD
Line commands:  INS - Insert Partition


ALTER TABLE : ID22279.TB22279


Sel   Part Limit Key Value
--- ------ ---------------------------------------------------------------------->
         1 'A'
         2 'B'
         3 'E'
******************************* END OF DB2 DATA *******************************
```

The partitions now listed on this panel reflect the removal of the partitions with limit key values of ‘C’ and ‘D’.

6. Issue the NEXT command.
7. Generate an ALT job

Adding LOB columns to an existing table

When you add a LOB column to a table, you can choose to explicitly create the related LOB objects or have Db2 implicitly create them for you. LOB objects include the LOB table space, the auxiliary table, and an index on the auxiliary table.

About this task

Db2 must implicitly create these LOB objects if the table already has LOB columns and any of the associated LOB objects were implicitly created.

You can choose to have Db2 implicitly create the LOB objects for the new LOB column in any of the following situations:

- The table does not have any existing LOB columns.

- Any existing LOB columns are being deleted by the same ALT command process that is adding the new LOB column.
- The table already has LOB columns and all associated LOB objects were explicitly created.

Procedure

To add LOB columns to an existing table:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify any selection criteria at the bottom, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel specify the ALT line command against the table to which you want to add LOB columns, and press Enter.
4. If the **Change Management Prompt (ADB2CMRO)** panel is displayed, specify whether you want to use the Change Management function to manage this change.
5. On the **ALTER Table (ADB27C)** panel, insert a new column by specifying the I line command and pressing Enter.

   If you want the new LOB column to be inserted at the beginning of the table, specify I in the empty row. Otherwise, the new column is inserted after the column where the I is specified.
6. In the **Col Type** column, specify the type of LOB that you want to create, such as `CLOB`.
7. Specify values in the other columns that contain a question mark (?), and press Enter.
8. Specify NEXT, and press Enter.
9. If the **Add LOB Objects Confirmation (ADB2CONF)** panel is displayed, specify whether you want to explicitly create the LOB objects.

   If this panel is not displayed, Db2 determined that it must implicitly create the LOB objects for you.

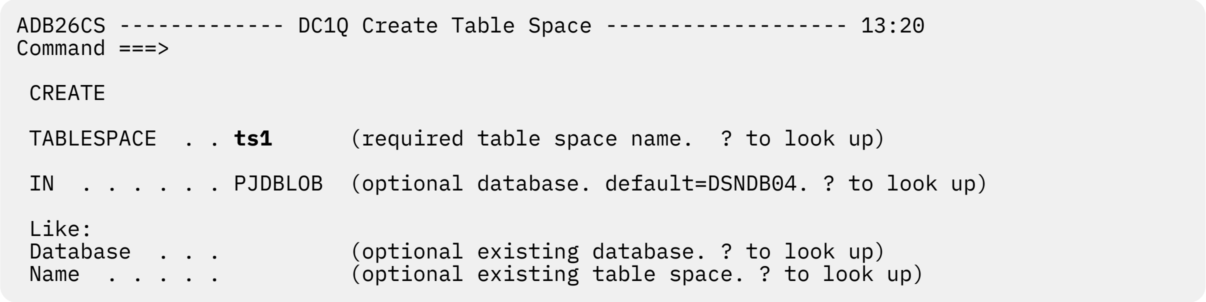
   ```
   ADB2CONF   DC1Q Alter Table - Add LOB Objects Confirmation   13:14


    You have added one or more LOB columns to the table.


    Select a choice
      1.  Create definitions for related LOB objects
      2.  Continue to have LOB objects created implicitly.
      3.  Cancel
   ```

   If you choose option 2, skip to step “14” on page 490.
10. Create the LOB table space:

    a) On the **Define Objects for New LOB Columns (ADBP7LOB)** panel, specify the CS line command next to the new LOB column, and press Enter to create the table space:

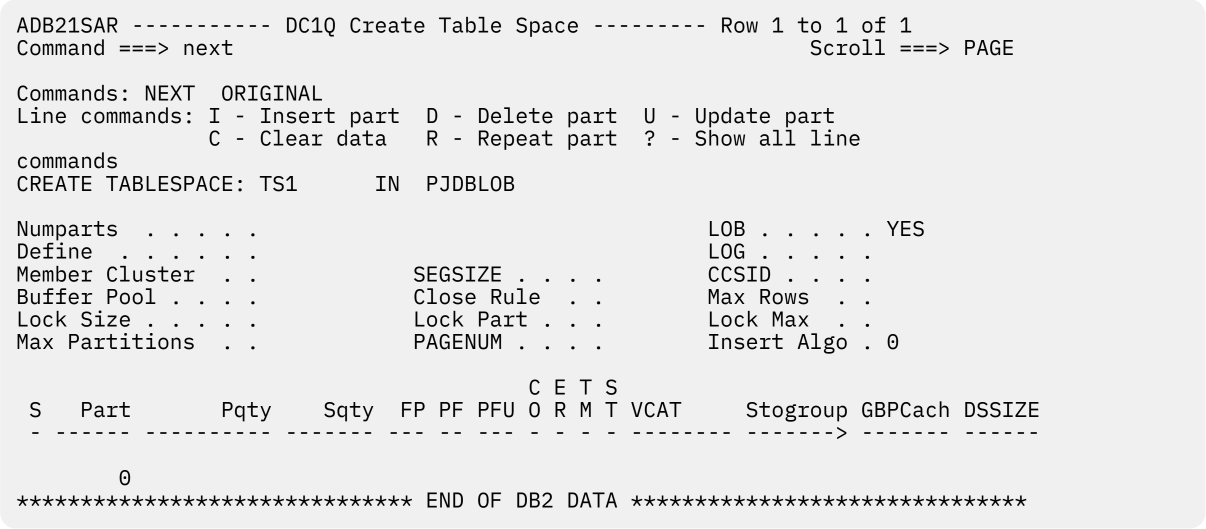
    ```
    ADBP7LOB   DC1Q Define Objects for New LOB Columns ---- Row 1 to 2 of 2
     Command ===>                                                  Scroll ===> PAGE

     Database     . : PJDBLOB
     Table schema . : MARLINO
     Table name . . : PJTBLBE2

     Commands: NEXT
     Line commands:
      CS - Create LOB TS  CA - Create aux table  CX - Create index on aux table

          New LOB                  LOB TS  Auxiliary-Table    Index    Index
     Sel  Column Name        Type   Name     Schema   Name     Schema   Name
          *                  *      *        *        *        *        *
     ---- -----------------> ------ -------- -------> -------> -------> ----------->
      CS  CL1                CLOB
          CL2                CLOB
     ****************************** END OF DB2 DATA ********************************
    ```

b) On the **Create Table Space (ADB26CS)** panel, Part 1, in the **TABLESPACE** field, specify the name of the table space, and press Enter:



```
ADB26CS ------------- DC1Q Create Table Space ------------------- 13:20
Command ===>

 CREATE

 TABLESPACE  . . ts1      (required table space name.  ? to look up)

 IN  . . . . . . PJDBLOB  (optional database. default=DSNDB04. ? to look up)

 Like:
 Database  . . .          (optional existing database. ? to look up)
 Name  . . . . .          (optional existing table space. ? to look up)
```

c) Optional: On the **Create Table Space (ADB21SAR)** panel, Part 2, specify values for the table space options:



```
ADB21SAR ----------- DC1Q Create Table Space --------- Row 1 to 1 of 1
Command ===> next                                             Scroll ===> PAGE

Commands: NEXT  ORIGINAL
Line commands: I - Insert part  D - Delete part  U - Update part
               C - Clear data   R - Repeat part  ? - Show all line
commands
CREATE TABLESPACE: TS1      IN  PJDBLOB

Numparts  . . . . .                                   LOB . . . . . YES
Define  . . . . . .                                   LOG . . . . .
Member Cluster  . .            SEGSIZE . . . .        CCSID . . . .
Buffer Pool . . . .            Close Rule  . .        Max Rows  . .
Lock Size . . . . .            Lock Part . . .        Lock Max  . .
Max Partitions  . .            PAGENUM . . . .        Insert Algo . 0

                                        C E T S
 S   Part       Pqty    Sqty  FP PF PFU O R M T VCAT     Stogroup GBPCach DSSIZE
 - ------ ---------- ------- --- -- --- - - - - -------- -------> ------- ------

       0
******************************* END OF DB2 DATA *******************************
```

Note: Insert Algo is displayed only if you are running Db2 12 for z/OS.

Restriction: You cannot specify values for the following options, because they are not allowed for LOB table spaces:

- Numparts
- Member Cluster
- Lock Size
- Max Partitions
- SEGSIZE
- PAGENUM
- CCSID
- Max Rows
- Free Page (FP)
- PCTFREE (PF)
- FOR UDPATE (PFU)
- Compress (CO)
- Trackmod (TM)

d) Specify NEXT, and press Enter.

The **Define Objects for New LOB Columns (ADBP7LOB)** panel is displayed again, and the new table space name is listed.

11. Create the auxiliary table:

    a) Specify the CA line command next to the new LOB column, and press Enter to create the auxiliary table.

    b) On the **Create Auxiliary Table (ADB26CA)** panel, specify the name of the auxiliary table, and press Enter:

```
ADB26CA ----------- DC1Q Create Auxiliary Table ----------------- 14:40
Command ===>

 CREATE AUXILIARY TABLE

 Schema . . . . . . MARLINO   >           (default is TS5775A)
 Name . . . . . . . aux1                > (? to look up)

 IN

 Database . . . . . PJDBLOB  (Default is DSNDB04. ? to look up)
 Table space  . . . TS1      (? to look up)

 STORES

 Table schema . . . MARLINO   >
 Table name . . . . PJTBLBB2            > (? to look up)

 COLUMN

 Column name  . . . CL1                 > (? to look up)

 PART . . . . . . .          (for which partition of base table)
```

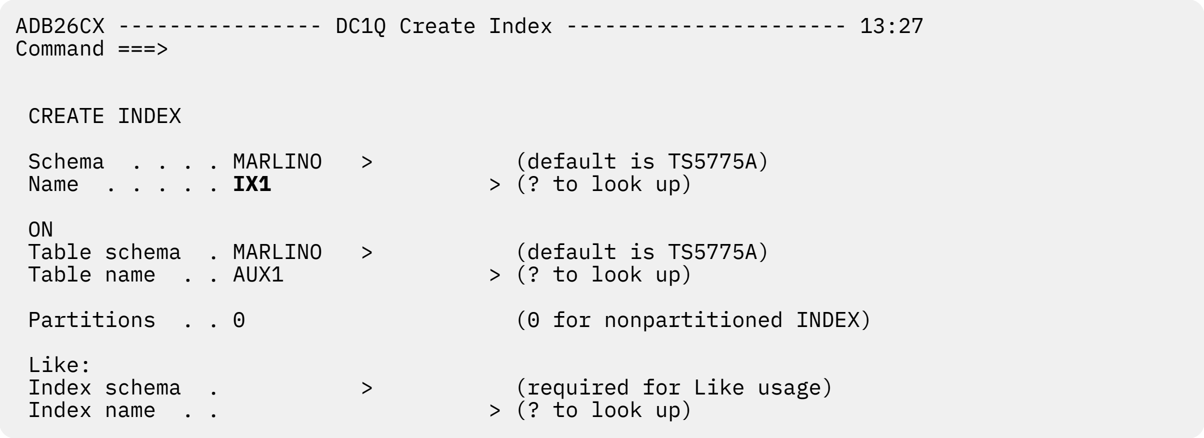
    The **Define Objects for New LOB Columns (ADBP7LOB)** panel is displayed again, and the new auxiliary table is listed:

```
ADBP7LOB   DC1Q Define Objects for New LOB Columns ---- Row 1 to 2 of 2
 Command ===>                                                Scroll ===> PAGE

 Database     . : PJDBLOB
 Table schema . : MARLINO
 Table name . . : PJTBLBE2

 Commands: NEXT
 Line commands:
  CS - Create LOB TS  CA - Create aux table  CX - Create index on aux table

      New LOB                   LOB TS  Auxiliary-Table    Index    Index
 Sel  Column Name        Type   Name     Schema   Name     Schema   Name
      *                  *      *        *        *        *        *
 ---- ------------------ ------ -------- -------- -------- -------- ------------
      CL1                CLOB   TS1      MARLINO  AUX1
      CL2                CLOB
 ****************************** END OF DB2 DATA ******************************
```

12. Create the index for the auxiliary table:

    a) Specify the CX line command next to the new LOB column, and press Enter to create the auxiliary index.

    b) On the **Create Index (ADB26CX)** panel, Part 1, specify a name for the index, and press Enter:

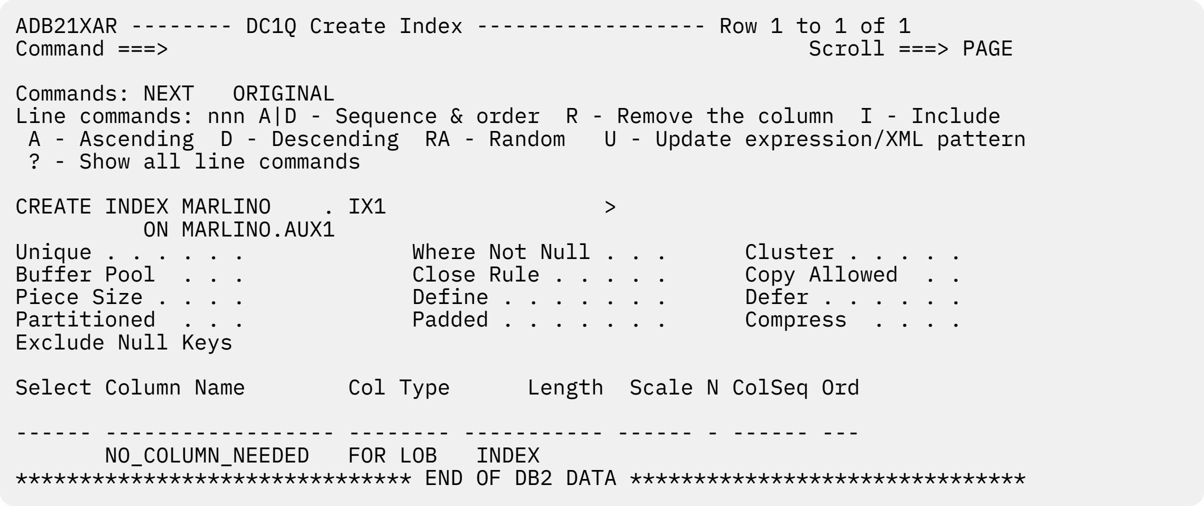


```
ADB26CX ---------------- DC1Q Create Index --------------------- 13:27
Command ===>


 CREATE INDEX

 Schema  . . . . MARLINO   >           (default is TS5775A)
 Name  . . . . . IX1                 > (? to look up)

 ON
 Table schema  . MARLINO   >           (default is TS5775A)
 Table name  . . AUX1                > (? to look up)

 Partitions  . . 0                     (0 for nonpartitioned INDEX)

 Like:
 Index schema  .           >           (required for Like usage)
 Index name  . .                     > (? to look up)
```

c) On the **Create Index (ADB21XAR)** panel, Part 2, specify values for the index options, specify NEXT, and press Enter.

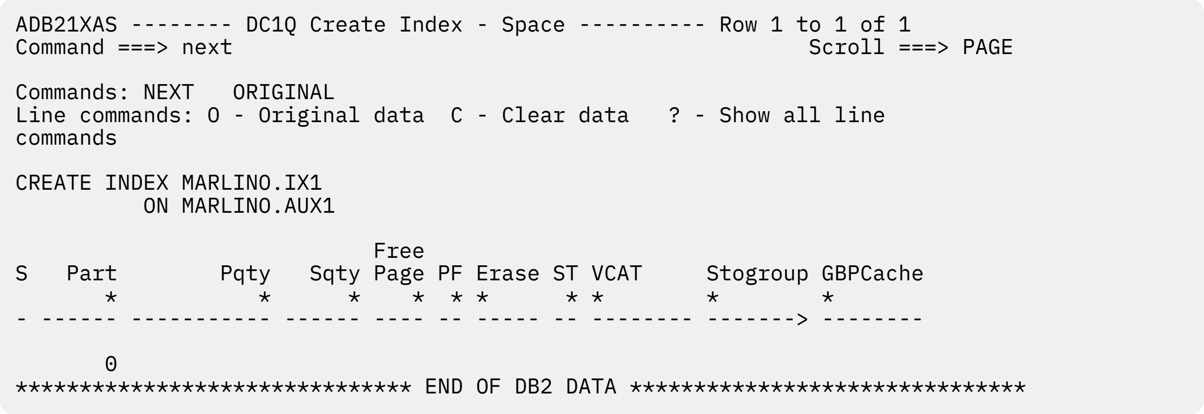
Restrictions:

- You cannot specify values for **Cluster** and **Defer**.
- Db2 ignores the values DEFINE NO and NOT PADDED.



```
ADB21XAR -------- DC1Q Create Index ------------------ Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Commands: NEXT   ORIGINAL
Line commands: nnn A|D - Sequence & order  R - Remove the column  I - Include
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern
 ? - Show all line commands

CREATE INDEX MARLINO    . IX1                 >
          ON MARLINO.AUX1
Unique . . . . . .            Where Not Null . . .      Cluster . . . . .
Buffer Pool  . . .            Close Rule . . . . .      Copy Allowed  . .
Piece Size . . . .            Define . . . . . . .      Defer . . . . . .
Partitioned  . . .            Padded . . . . . . .      Compress  . . . .
Exclude Null Keys

Select Column Name        Col Type      Length  Scale N ColSeq Ord

------ ------------------ -------- ----------- ------ - ------ ---
       NO_COLUMN_NEEDED   FOR LOB   INDEX
******************************* END OF DB2 DATA *******************************
```

d) On the **Create Index (ADB21XAS)** panel, Part 3, specify NEXT, and press Enter:



```
ADB21XAS -------- DC1Q Create Index - Space ---------- Row 1 to 1 of 1
Command ===> next                                             Scroll ===> PAGE

Commands: NEXT   ORIGINAL
Line commands: O - Original data  C - Clear data   ? - Show all line
commands

CREATE INDEX MARLINO.IX1
          ON MARLINO.AUX1

                            Free
S   Part        Pqty   Sqty Page PF Erase ST VCAT     Stogroup GBPCache
       *           *      *    *  * *      * *        *        *
- ------ ----------- ------ ---- -- ----- -- -------- -------> --------

       0
******************************* END OF DB2 DATA *******************************
```

The **Define Objects for New LOB Columns (ADBP7LOB)** panel panel is displayed again, and the new index is listed.

13. Optional: To create objects for any other new LOB columns, repeat steps "10" on page 486 through "12" on page 488 as needed.

14. After you create objects for all the new LOB columns, specify NEXT, and press Enter.

    The **Alter Objects (ADB27CA)** panel is displayed.
15. Generate an ALT job

Altering indexes

You can rename an index, change index attributes, and change the columns that are indexed. Depending on the change that you want to make, use either the AL or ALT line command.

About this task

Use the AL line command to make changes that are supported by the ALTER INDEX statement.

Use the ALT line command to alter an index when the changes are more complex and intrusive and cannot be accomplished with an ALTER INDEX statement. An intrusive alter is one in which the objects have to be dropped and re-created, such as inserting a column in the middle of a table, dropping a column, renaming a column, or changing the attributes of a column. When you specify your index redefinition parameters, you can choose to save your request to a work statement list.

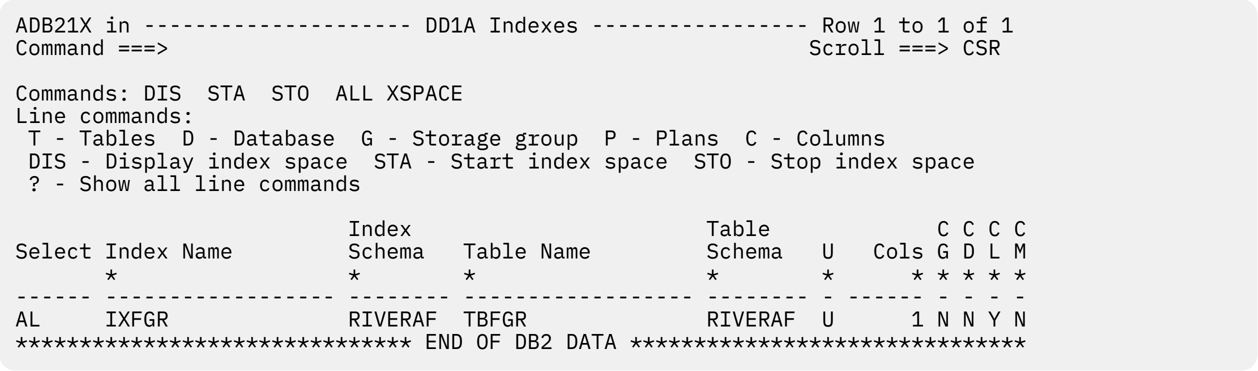
- “Altering an index by using the AL line command” on page 490
- “Altering an index by using the ALT line command” on page 491

Altering an index by using the AL line command

Procedure

To alter an index by using the AL line command:

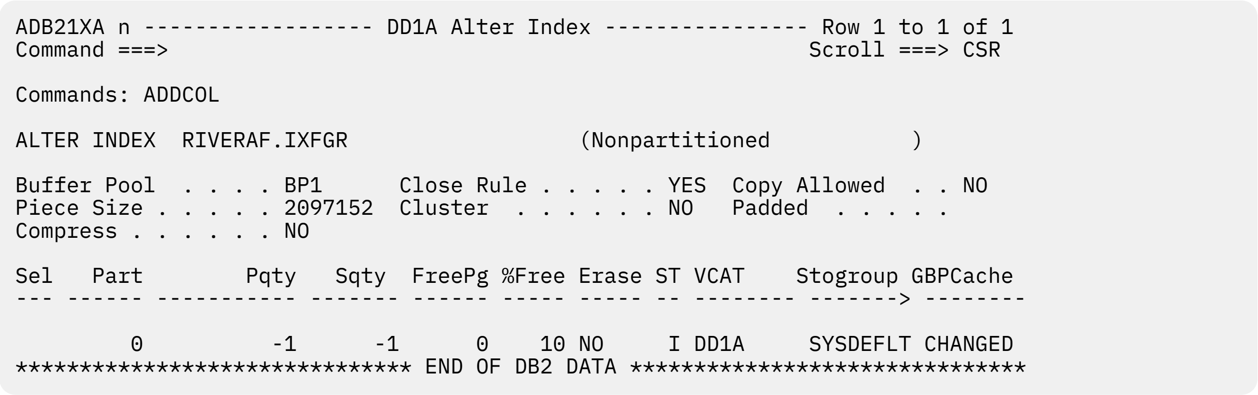
1. On the **DB2 Administration Menu (ADB2)** panel, select option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, select option X, and press Enter.
3. On the **Indexes (ADB21X)** panel, enter the AL line command against the index that you want to alter.



```
ADB21X in --------------------- DD1A Indexes ----------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Commands: DIS  STA  STO  ALL XSPACE
Line commands:
 T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
 DIS - Display index space  STA - Start index space  STO - Stop index space
 ? - Show all line commands

                          Index                        Table              C C C C
Select Index Name         Schema   Table Name          Schema   U   Cols G D L M
       *                  *        *                   *        *      * * * * *
------ ------------------ -------- ------------------ -------- - ------ - - - -
AL     IXFGR              RIVERAF  TBFGR              RIVERAF  U      1 N N Y N
******************************* END OF DB2 DATA *******************************
```

*Figure 255. **Indexes (ADB21X)** panel*

4. On the **Alter Index (ADB21XA)** panel, make changes to any index attributes, and press Enter.



```
ADB21XA n ------------------ DD1A Alter Index ---------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Commands: ADDCOL

ALTER INDEX  RIVERAF.IXFGR                  (Nonpartitioned          )

Buffer Pool  . . . . BP1      Close Rule . . . . . YES  Copy Allowed  . . NO
Piece Size . . . . . 2097152  Cluster  . . . . . . NO   Padded  . . . . .
Compress . . . . . . NO

Sel   Part        Pqty   Sqty  FreePg %Free Erase ST VCAT    Stogroup GBPCache
--- ------ ----------- ------- ------ ----- ----- -- -------- -------> --------

         0          -1      -1      0    10 NO     I DD1A     SYSDEFLT CHANGED
******************************* END OF DB2 DATA *******************************
```

*Figure 256. **Alter Index (ADB21XA)** panel*

For a partitioning index, a detail line is displayed for each partition. You can alter any partition by updating the available attribute, such as `%Free`. To apply the same change to all partitions of the index, provide a value in the All Parts row.

After you press Enter, Db2 Admin Tool runs the SQL ALTER INDEX statement. Depending on your prompt options, you might need to first confirm that you want to execute the ALTER statement.

To change certain parameters, you must stop and restart the associated object. In these cases, Db2 Admin Tool issues a STOP table space or STOP index (or partition) command and checks that the object is in a fully-stopped state. If stopped, it issues an ALTER INDEX statement, followed by a START command. If the object is not in a fully-stopped state, the **STOP Check - Action** panel panel prompts you to perform one of the following actions:

- Check again
- Issue the ALTER
- Cancel the operation

If an object is not stopped when the ALTER TABLESPACE statement is run (for example, if others are holding locks on the object), a -626 SQLCODE is displayed.

Related tasks

"Changing Db2 Admin Tool prompt options" on page 243

Altering an index by using the ALT line command

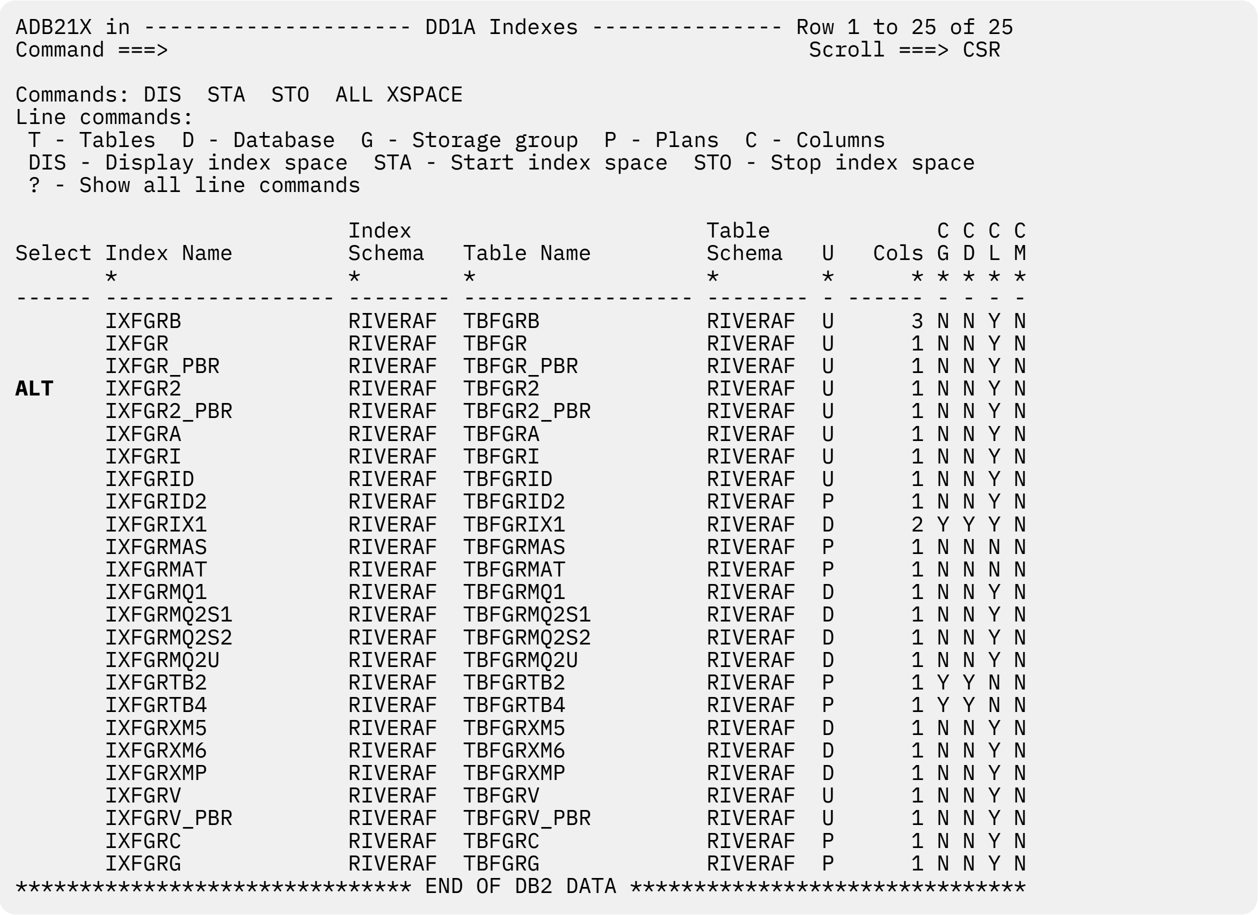
About this task

Using the ALT command to alter an index or partitioning index is called *redefining* the index.

Procedure

To redefine an index or a partitioning index:

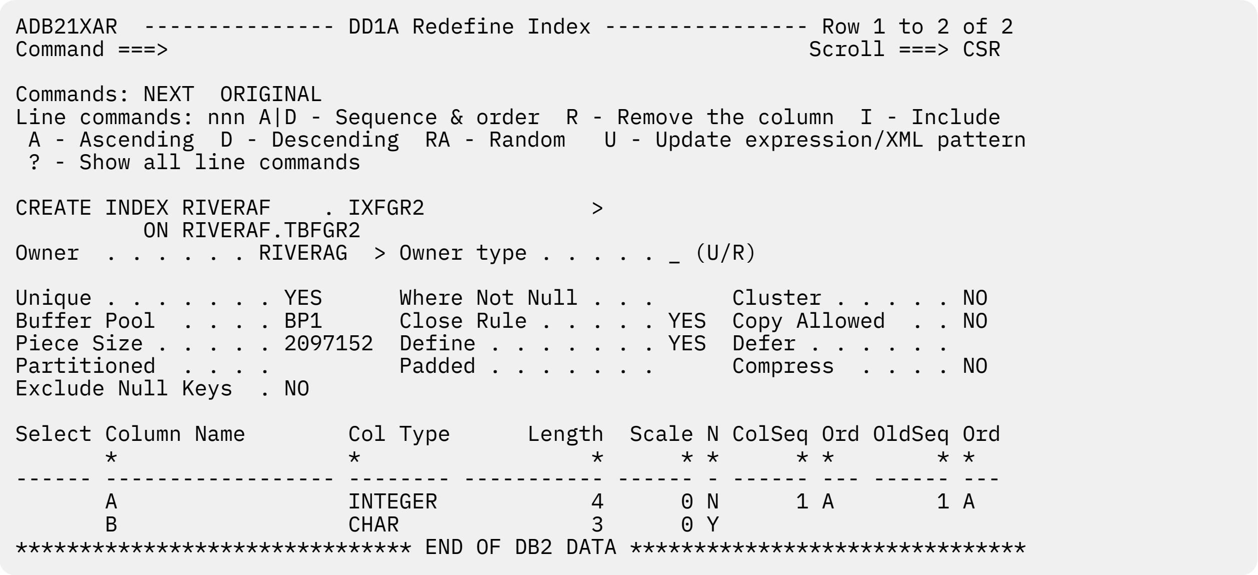
1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option X and optionally any filtering criteria at the bottom of the panel, and press Enter.
3. On the **Indexes (ADB21X)** panel, issue the ALT line command next to the index that you want to redefine, and press Enter:



```
ADB21X in -------------------- DD1A Indexes --------------- Row 1 to 25 of 25
Command ===>                                                  Scroll ===> CSR

Commands: DIS  STA  STO  ALL XSPACE
Line commands:
 T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
 DIS - Display index space  STA - Start index space  STO - Stop index space
 ? - Show all line commands

                          Index                        Table              C C C C
Select Index Name         Schema   Table Name          Schema   U   Cols G D L M
       *                  *        *                   *        *      * * * * *
------ ------------------ -------- ------------------ -------- - ------ - - - -
       IXFGRB             RIVERAF  TBFGRB             RIVERAF  U      3 N N Y N
       IXFGR              RIVERAF  TBFGR              RIVERAF  U      1 N N Y N
       IXFGR_PBR          RIVERAF  TBFGR_PBR          RIVERAF  U      1 N N Y N
ALT    IXFGR2             RIVERAF  TBFGR2             RIVERAF  U      1 N N Y N
       IXFGR2_PBR         RIVERAF  TBFGR2_PBR         RIVERAF  U      1 N N Y N
       IXFGRA             RIVERAF  TBFGRA             RIVERAF  U      1 N N Y N
       IXFGRI             RIVERAF  TBFGRI             RIVERAF  U      1 N N Y N
       IXFGRID            RIVERAF  TBFGRID            RIVERAF  U      1 N N Y N
       IXFGRID2           RIVERAF  TBFGRID2           RIVERAF  P      1 N N Y N
       IXFGRIX1           RIVERAF  TBFGRIX1           RIVERAF  D      2 Y Y Y N
       IXFGRMAS           RIVERAF  TBFGRMAS           RIVERAF  P      1 N N N N
       IXFGRMAT           RIVERAF  TBFGRMAT           RIVERAF  P      1 N N N N
       IXFGRMQ1           RIVERAF  TBFGRMQ1           RIVERAF  D      1 N N Y N
       IXFGRMQ2S1         RIVERAF  TBFGRMQ2S1         RIVERAF  D      1 N N Y N
       IXFGRMQ2S2         RIVERAF  TBFGRMQ2S2         RIVERAF  D      1 N N Y N
       IXFGRMQ2U          RIVERAF  TBFGRMQ2U          RIVERAF  D      1 N N Y N
       IXFGRTB2           RIVERAF  TBFGRTB2           RIVERAF  P      1 Y Y N N
       IXFGRTB4           RIVERAF  TBFGRTB4           RIVERAF  P      1 Y Y N N
       IXFGRXM5           RIVERAF  TBFGRXM5           RIVERAF  D      1 N N Y N
       IXFGRXM6           RIVERAF  TBFGRXM6           RIVERAF  D      1 N N Y N
       IXFGRXMP           RIVERAF  TBFGRXMP           RIVERAF  D      1 N N Y N
       IXFGRV             RIVERAF  TBFGRV             RIVERAF  U      1 N N Y N
       IXFGRV_PBR         RIVERAF  TBFGRV_PBR         RIVERAF  U      1 N N Y N
       IXFGRC             RIVERAF  TBFGRC             RIVERAF  P      1 N N Y N
       IXFGRG             RIVERAF  TBFGRG             RIVERAF  P      1 N N Y N
******************************* END OF DB2 DATA ******************************
```

Figure 257. ***Indexes (ADB21X)*** *panel*

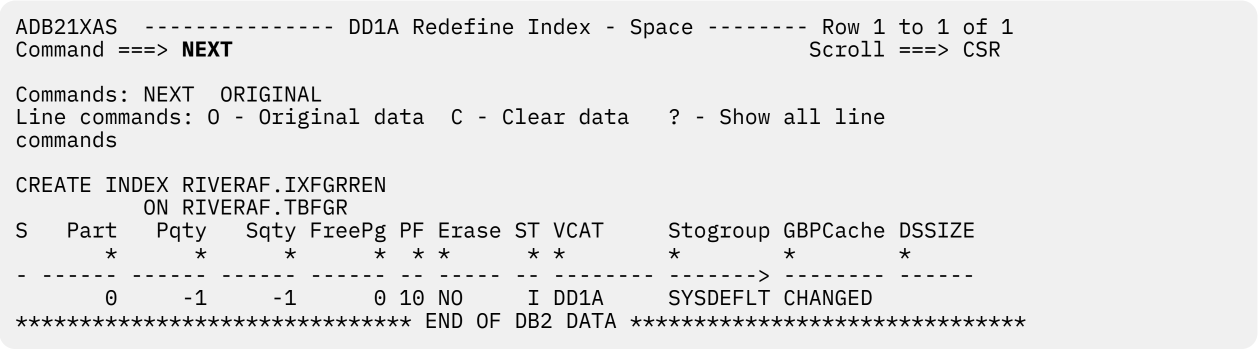
4. On the **Redefine Index (ADB21XAR)** panel, alter any index attributes and press Enter:



```
ADB21XAR  --------------- DD1A Redefine Index ---------------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: nnn A|D - Sequence & order  R - Remove the column  I - Include
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern
 ? - Show all line commands

CREATE INDEX RIVERAF    . IXFGR2            >
          ON RIVERAF.TBFGR2
Owner  . . . . . . RIVERAG  > Owner type . . . . . _ (U/R)

Unique . . . . . . . YES      Where Not Null . . .      Cluster . . . . . NO
Buffer Pool  . . . . BP1      Close Rule . . . . . YES  Copy Allowed  . . NO
Piece Size . . . . . 2097152  Define . . . . . . . YES  Defer . . . . . .
Partitioned  . . . .          Padded . . . . . . .      Compress  . . . . NO
Exclude Null Keys  . NO

Select Column Name        Col Type      Length  Scale N ColSeq Ord OldSeq Ord
       *                  *                  *      * *      * *        * *
------ ------------------ -------- ----------- ------ - ------ --- ------ ---
       A                  INTEGER            4      0 N      1 A        1 A
       B                  CHAR               3      0 Y
******************************* END OF DB2 DATA ******************************
```

Figure 258. ***Redefine Index (ADB21XAR)*** *panel*

5. Issue the primary command NEXT, and press Enter.
6. On the **Redefine Index - Space (ADB21XAS)** panel, issue NEXT again, and press Enter:



```
ADB21XAS  --------------- DD1A Redefine Index - Space -------- Row 1 to 1 of 1
Command ===> NEXT                                             Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: O - Original data  C - Clear data   ? - Show all line
commands

CREATE INDEX RIVERAF.IXFGRREN
          ON RIVERAF.TBFGR
S   Part   Pqty   Sqty FreePg PF Erase ST VCAT     Stogroup GBPCache DSSIZE
       *      *      *      *  * *      * *        *        *        *
- ------ ------ ------ ------ -- ----- -- -------- -------> -------- ------
       0     -1     -1      0 10 NO     I DD1A     SYSDEFLT CHANGED
******************************* END OF DB2 DATA *******************************
```

Figure 259. ***Redefine Index - Space (ADB21XAS)*** *panel*

7. Generate an ALT job.

Example

The following topics show specific examples of altering indexes:

- "Renaming indexes" on page 493
- "Excluding null keys from an index" on page 494

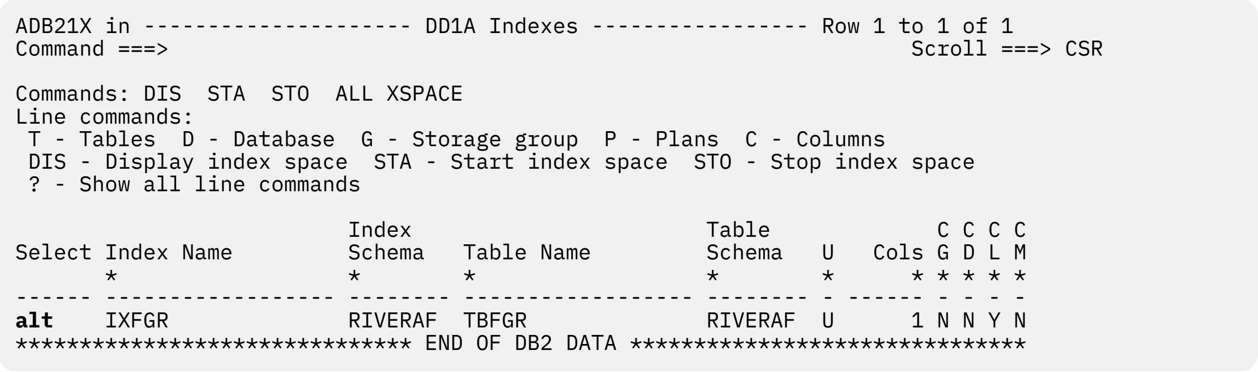
Renaming indexes

You can rename an index if you are the owner of the index or have the DBA privilege on the database.

Procedure

To rename an index:

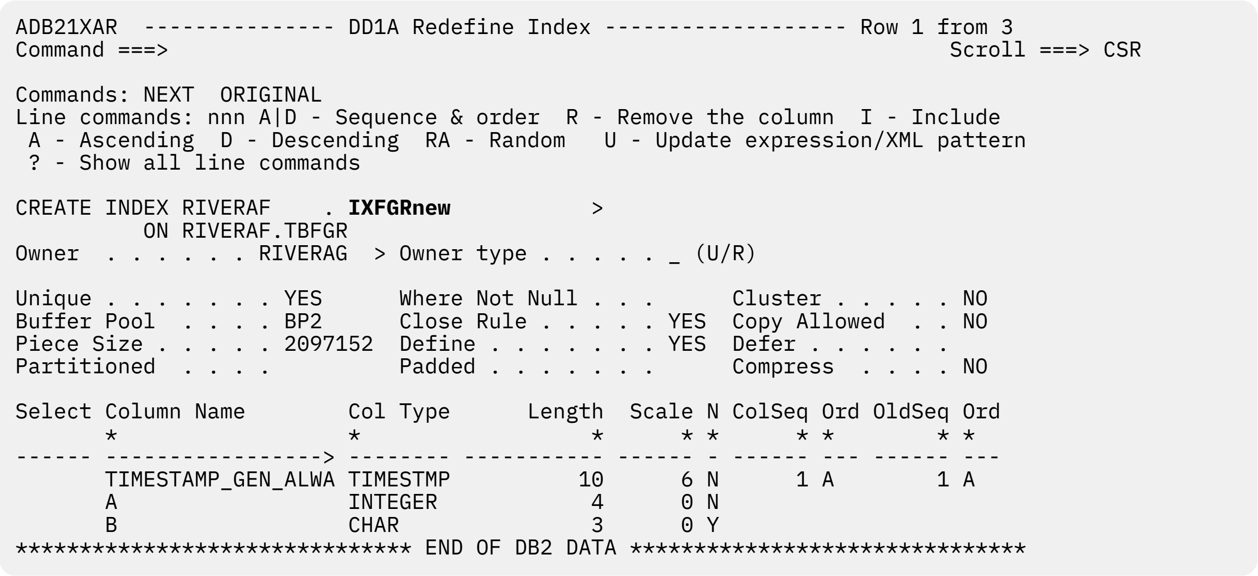
1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option X and optionally any filtering criteria at the bottom of the panel, and press Enter.
3. On the **Indexes (ADB21X)** panel, issue the ALT line command next to the index that you want to rename, and press Enter:



```
ADB21X in --------------------- DD1A Indexes ----------------- Row 1 to 1 of 1
Command ===>                                                          Scroll ===> CSR

Commands: DIS  STA  STO  ALL XSPACE
Line commands:
 T - Tables  D - Database  G - Storage group  P - Plans  C - Columns
 DIS - Display index space  STA - Start index space  STO - Stop index space
 ? - Show all line commands

                          Index                       Table             C C C C
Select Index Name         Schema   Table Name         Schema   U   Cols G D L M
       *                  *        *                  *        *      * * * * *
------ ------------------ -------- ------------------ -------- - ------ - - - -
alt    IXFGR              RIVERAF  TBFGR              RIVERAF  U      1 N N Y N
******************************* END OF DB2 DATA *******************************
```

Figure 260. ***Indexes (ADB21X)*** *panel*

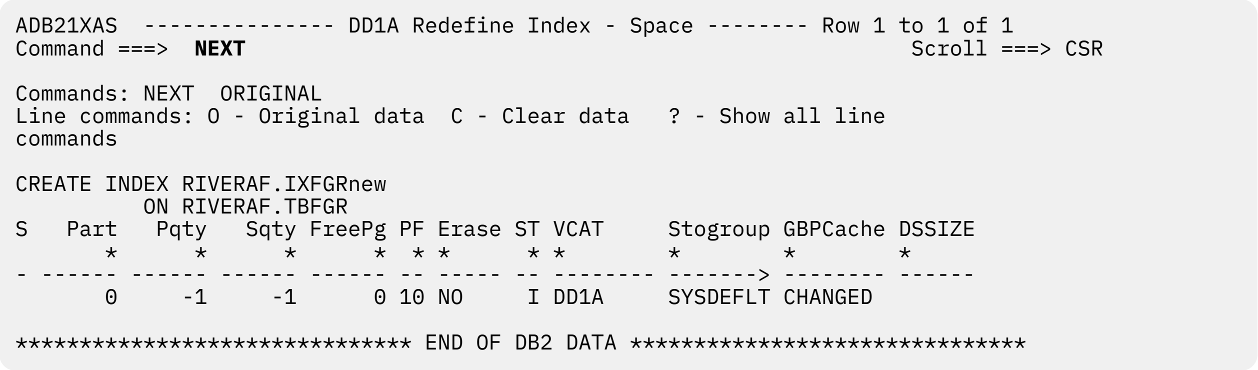
4. On the **Redefine Index (ADB21XAR)** panel, in the **CREATE INDEX** field, type over the original index name with the new name.



```
ADB21XAR  --------------- DD1A Redefine Index ------------------- Row 1 from 3
Command ===>                                                         Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: nnn A|D - Sequence & order  R - Remove the column  I - Include
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern
 ? - Show all line commands

CREATE INDEX RIVERAF    . IXFGRnew           >
          ON RIVERAF.TBFGR
Owner  . . . . . . RIVERAG  > Owner type . . . . . _ (U/R)

Unique . . . . . . . YES      Where Not Null . . .      Cluster . . . . . NO
Buffer Pool  . . . . BP2      Close Rule . . . . . YES  Copy Allowed  . . NO
Piece Size . . . . . 2097152  Define . . . . . . . YES  Defer . . . . . .
Partitioned  . . . .          Padded . . . . . . .      Compress  . . . . NO

Select Column Name        Col Type      Length  Scale N ColSeq Ord OldSeq Ord
       *                  *                  *      * *      * *        * *
------ ----------------> -------- ---------- ------ - ------ --- ------ ---
       TIMESTAMP_GEN_ALWA TIMESTMP          10      6 N      1 A        1 A
       A                  INTEGER            4      0 N
       B                  CHAR               3      0 Y
******************************* END OF DB2 DATA *******************************
```

Figure 261. ***Redefine Index (ADB21XAR)*** *panel*

5. Issue the primary command NEXT, and press Enter.
6. On the **Redefine Index - Space (ADB21XAS)** panel, issue NEXT again, and press Enter:



```
ADB21XAS  --------------- DD1A Redefine Index - Space -------- Row 1 to 1 of 1
Command ===>  NEXT                                                      Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: O - Original data  C - Clear data   ? - Show all line
commands

CREATE INDEX RIVERAF.IXFGRnew
          ON RIVERAF.TBFGR
S   Part   Pqty   Sqty FreePg PF Erase ST VCAT     Stogroup GBPCache DSSIZE
       *      *      *      *  * *      * *        *        *        *
- ------ ------ ------ ------ -- ----- -- -------- -------> -------- ------
       0     -1     -1      0 10 NO     I DD1A     SYSDEFLT CHANGED

******************************* END OF DB2 DATA *******************************
```

Figure 262. ***Redefine Index - Space (ADB21XAS)*** *panel*

7. Generate an ALT job.

Excluding null keys from an index

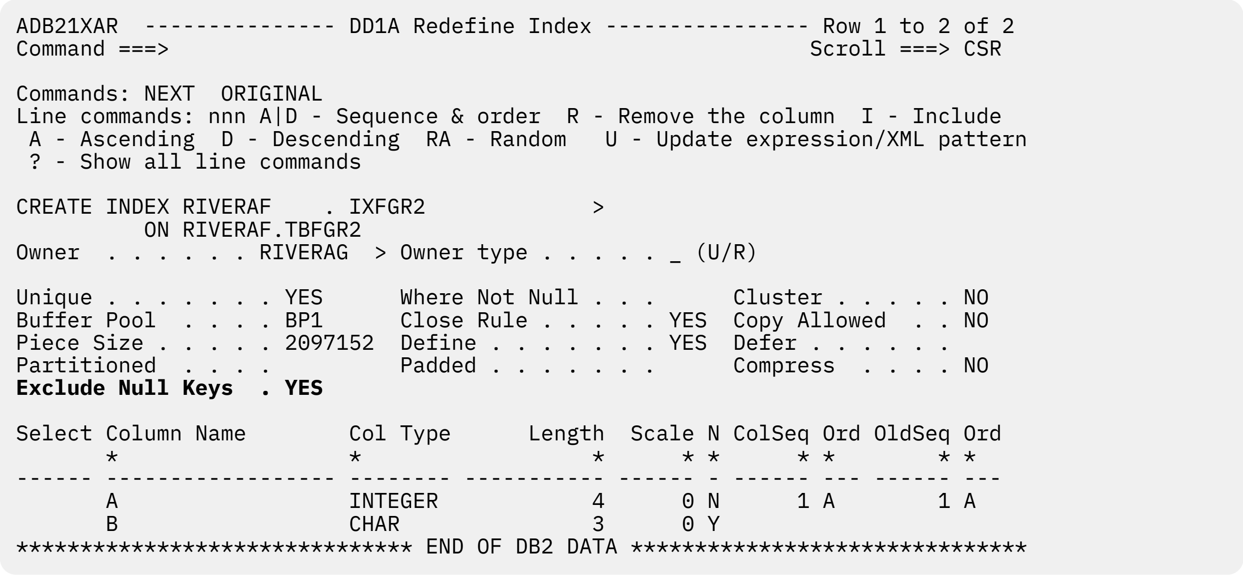
To save index space and to improve INSERT and query performance, you can redefine an index so that it no longer contains null keys.

About this task

In this example, the index IXFGR2 was originally created with null keys.

Procedure

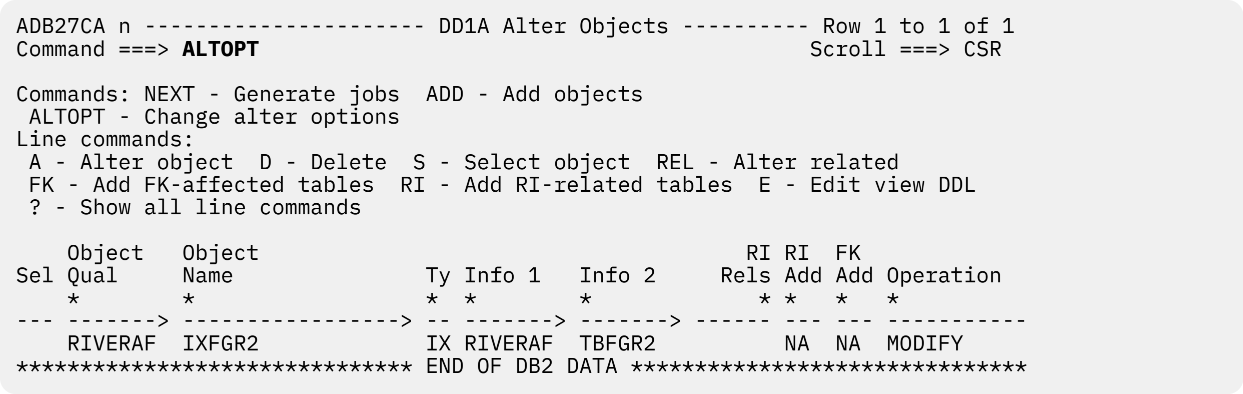
1. On the **DB2 Administration Menu (ADB2)** panel, select option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, select option X, and press Enter.
3. On the **Indexes (ADB21X)** panel, issue the ALT line command next to the index that you want to redefine.
4. On the **Redefine Index (ADB21XAR)** panel, type YES in the **Exclude Null Keys** field and press Enter:



```
ADB21XAR  --------------- DD1A Redefine Index ---------------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> CSR

Commands: NEXT  ORIGINAL
Line commands: nnn A|D - Sequence & order  R - Remove the column  I - Include
 A - Ascending  D - Descending  RA - Random   U - Update expression/XML pattern
 ? - Show all line commands

CREATE INDEX RIVERAF    . IXFGR2             >
          ON RIVERAF.TBFGR2
Owner  . . . . . . RIVERAG  > Owner type . . . . . _ (U/R)

Unique . . . . . . . YES      Where Not Null . . .      Cluster . . . . . NO
Buffer Pool  . . . . BP1      Close Rule . . . . . YES  Copy Allowed  . . NO
Piece Size . . . . . 2097152  Define . . . . . . . YES  Defer . . . . . .
Partitioned  . . . .          Padded . . . . . . .      Compress  . . . . NO
Exclude Null Keys  . YES

Select Column Name        Col Type      Length  Scale N ColSeq Ord OldSeq Ord
       *                  *                  *      * *      * *        * *
------ ------------------ -------- ----------- ------ - ------ --- ------ ---
       A                  INTEGER            4      0 N      1 A        1 A
       B                  CHAR               3      0 Y
******************************* END OF DB2 DATA *******************************
```

*Figure 263. **Redefine Index (ADB21XAR)** panel - Redefining Exclude Null Keys attribute*

5. Issue the NEXT primary command, and press Enter.
6. On the **Redefine Index - Space (ADB21XAS)** panel, issue NEXT again, and press Enter.
7. On the **Alter Objects (ADB27CA)** panel, issue the ALTOPT command, and press Enter:



```
ADB27CA n ---------------------- DD1A Alter Objects ---------- Row 1 to 1 of 1
Command ===> ALTOPT                                           Scroll ===> CSR

Commands: NEXT - Generate jobs  ADD - Add objects
 ALTOPT - Change alter options
Line commands:
 A - Alter object  D - Delete  S - Select object  REL - Alter related
 FK - Add FK-affected tables  RI - Add RI-related tables  E - Edit view DDL
 ? - Show all line commands

    Object   Object                                  RI RI  FK
Sel Qual     Name              Ty Info 1   Info 2     Rels Add Add Operation
    *        *                 *  *        *             * *   *   *
--- -------> -----------------> -- -------> -------> ------ --- --- -----------
    RIVERAF  IXFGR2            IX RIVERAF  TBFGR2          NA  NA  MODIFY
******************************* END OF DB2 DATA *******************************
```

*Figure 264. **Alter Objects (ADB27CA)** panel*

8. On the **ALTER Options Menu (ADBP7OP)** panel, select option 1, and press Enter.
9. On the **ALTER Analysis Options (ADBP7P)** panel, type YES in the **Perform analysis in batch** field.
10. Exit back to the **Alter Objects (ADB27CA)** panel.
11. Generate an ALT job.

Altering triggers

You can make changes to existing triggers, such as modifying the attributes and versions.

About this task

Use the AL line command to make changes that can made by an ALTER TRIGGER statement. Use the ALT command to make other changes.

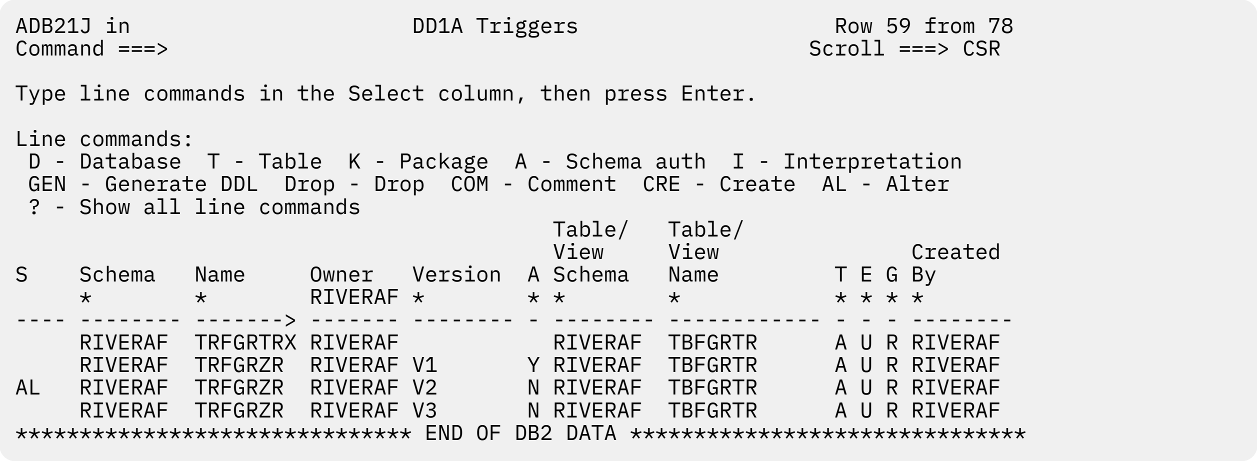
Procedure

To alter a trigger:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1, and press Enter.

2. On the **System Catalog (ADB21)** panel, select option J, and press Enter.
3. On the **Triggers (ADB21J)** panel, specify the AL or ALT line command.

   This example assumes that you want to make a simple change that can be achieved by using an ALTER statement and therefore uses the AL line command:



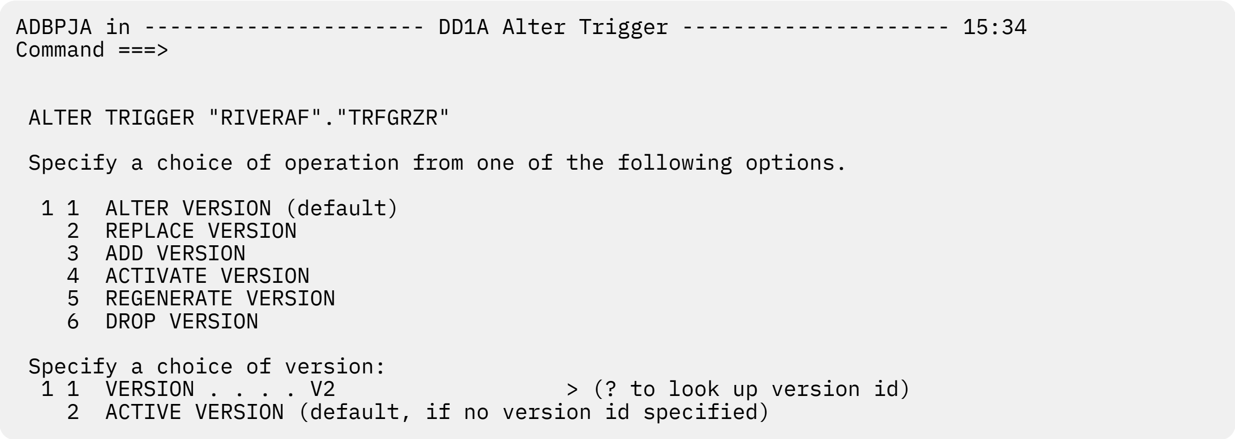
```
ADB21J in                      DD1A Triggers                   Row 59 from 78
Command ===>                                                 Scroll ===> CSR

Type line commands in the Select column, then press Enter.

Line commands:
 D - Database  T - Table  K - Package  A - Schema auth  I - Interpretation
 GEN - Generate DDL  Drop - Drop  COM - Comment  CRE - Create  AL - Alter
 ? - Show all line commands
                                          Table/   Table/
                                          View     View                Created
S    Schema   Name     Owner   Version  A Schema   Name         T E G By
     *        *        RIVERAF *        * *        *            * * * *
---- -------- -------> ------- -------- - -------- ------------ - - - --------
     RIVERAF  TRFGRTRX RIVERAF            RIVERAF  TBFGRTR      A U R RIVERAF
     RIVERAF  TRFGRZR  RIVERAF V1       Y RIVERAF  TBFGRTR      A U R RIVERAF
AL   RIVERAF  TRFGRZR  RIVERAF V2       N RIVERAF  TBFGRTR      A U R RIVERAF
     RIVERAF  TRFGRZR  RIVERAF V3       N RIVERAF  TBFGRTR      A U R RIVERAF
******************************* END OF DB2 DATA *******************************
```

Figure 265. Triggers (ADB21J)

   The remaining instructions apply to the AL line command. For instructions on how to use the ALT line command, see "ALT (Alter) function" on page 447.

   One of the following versions of the **Alter Trigger (ADBPJA)** panel is displayed:

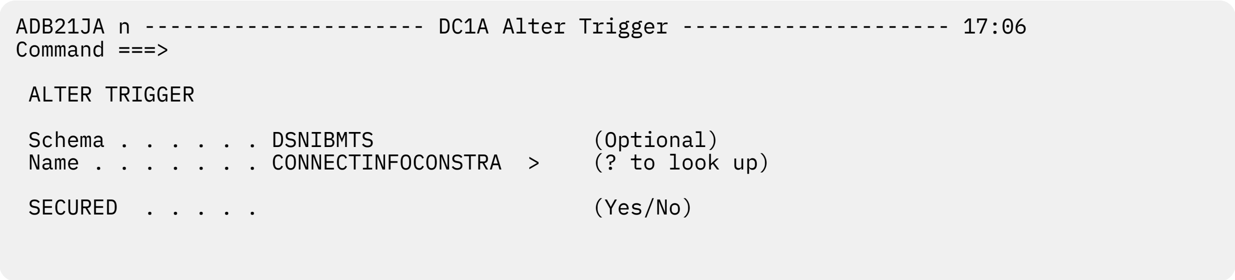
   - Options for editing trigger versions:



```
ADBPJA in ---------------------- DD1A Alter Trigger --------------------- 15:34
Command ===>


 ALTER TRIGGER "RIVERAF"."TRFGRZR"

 Specify a choice of operation from one of the following options.

  1 1  ALTER VERSION (default)
    2  REPLACE VERSION
    3  ADD VERSION
    4  ACTIVATE VERSION
    5  REGENERATE VERSION
    6  DROP VERSION

 Specify a choice of version:
  1 1  VERSION . . . . V2                  > (? to look up version id)
    2  ACTIVE VERSION (default, if no version id specified)
```

*Figure 266. **Alter Trigger (ADBPJA)** panel*

   - Basic options:



```
ADB21JA n ---------------------- DC1A Alter Trigger --------------------- 17:06
Command ===>

 ALTER TRIGGER

 Schema . . . . . . DSNIBMTS                 (Optional)
 Name . . . . . . . CONNECTINFOCONSTRA  >    (? to look up)

 SECURED  . . . . .                          (Yes/No)
```

*Figure 267. **Alter Trigger (ADBPJA)** panel*

4. On the **Alter Trigger (ADBPJA)** panel specify the appropriate options, and press Enter.

5. If the **Alter Trigger (ADBPJA1)** panel is displayed, specify your changes, and issue the NEXT command.

   Db2 Admin Tool issues an ALTER TRIGGER statement to make the changes.

Altering views

You cannot use an SQL ALTER statement to change a view. Therefore, to make changes to a view, Db2 Admin Tool generates a set of jobs that drop the view and then re-create it.

Procedure

To alter views:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify any selection criteria at the bottom, specify option V, and press Enter.
3. On the **Views (ADB21VV)** panel, specify one of the following line commands next to the view that you want to change:

   REN
      Specify this line command to rename the view. Specify the new name on the **Rename VIEW (ADB21VR)** panel. Press Enter, and skip to step "5" on page 497.

   ALT
      Specify this line command for all other changes. Panel ADB27CAA might appear briefly while the definition of the view is being retrieved. An SQL CREATE VIEW statement for the view is displayed in an ISPF Edit Session.
4. Edit the CREATE VIEW statement to make the changes that you want, and press PF3.

   The **Alter Objects (ADB27CA)** panel lists the view. The **Operation** column shows the action MODIFY. If you did not change the CREATE VIEW statement, no operation is listed.
5. Generate an ALT job

Related concepts

"ALT (Alter) function" on page 447
When you use the ALT line command to change an object, you invoke the ALT function of Db2 Admin Tool. The ALT function first analyzes the change to determine what actions need to be taken to implement the change. You can choose whether you want this analysis to be done in batch or online.

"AL (Alter) function" on page 447
When you use the AL line command to change an object, you invoke the AL function of Db2 Admin Tool. The AL function makes the requested object changes by executing an SQL ALTER statement.

Related reference

"Option V. Views" on page 185
The **Views (ADB21VV)** panel displays information about views, including whether the view uses multiple tables in different databases or table spaces. Alternatively, you can see information about views on the **Tables, Views, and Aliases (ADB21T)** panel. However, that panel does not show whether multiple databases or table spaces are used by the view.

Altering foreign keys

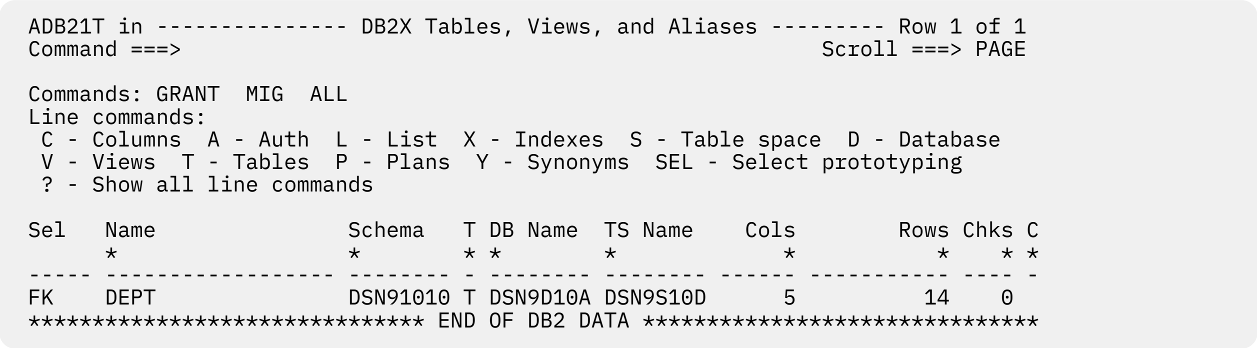
To make changes to foreign key attributes, you must use the ALT line command.

Procedure

To alter foreign keys:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify any selection criteria at the bottom to find the table with the foreign keys, specify option T, and press Enter.

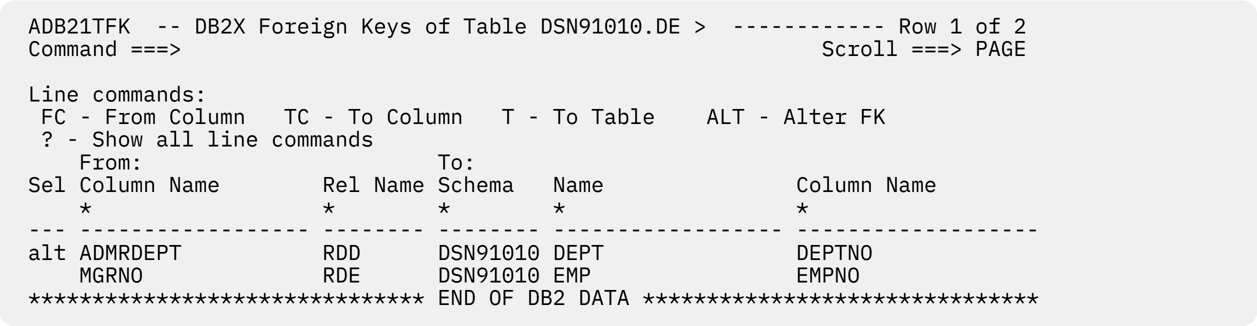
3. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the FK line command against a table to display the foreign keys for that table.



```
ADB21T in --------------- DB2X Tables, Views, and Aliases --------- Row 1 of 1
Command ===>                                                  Scroll ===> PAGE

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

Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C
      *                  *        * *        *             *           *    * *
----- ------------------ -------- - -------- -------- ------ ----------- ---- -
FK    DEPT               DSN91010 T DSN9D10A DSN9S10D      5          14    0
******************************* END OF DB2 DATA *******************************
```

*Figure 268. **Tables, Views, and Aliases (ADB21T)** panel*

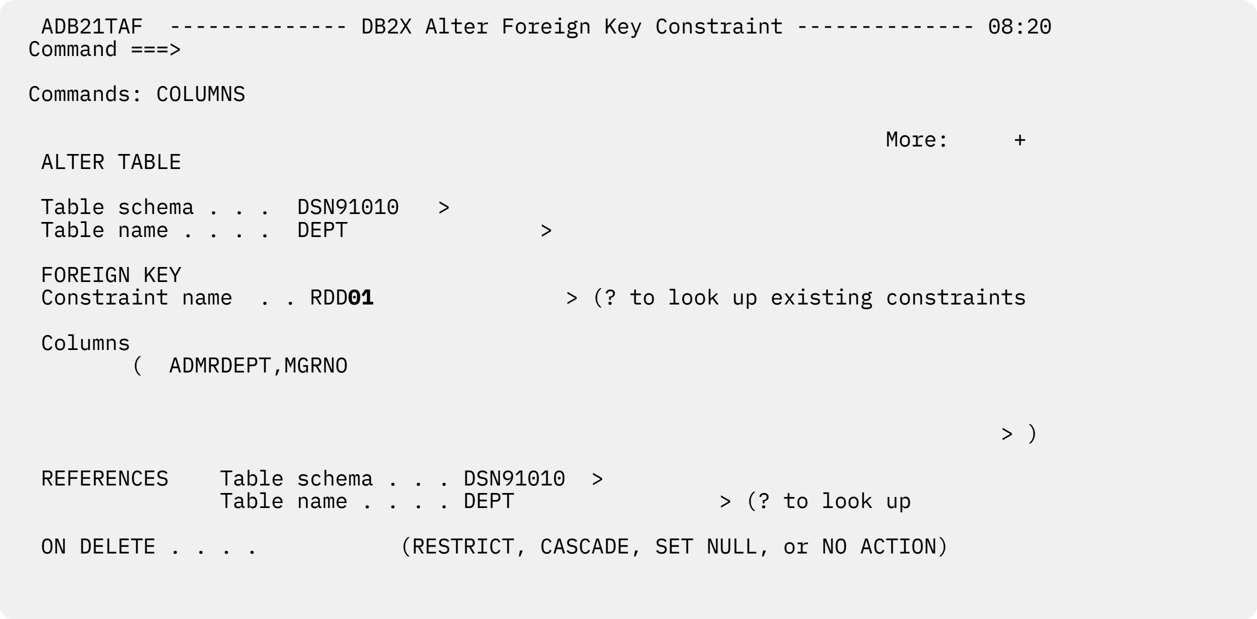
4. On the **Foreign Keys of Table (ADB21TFK)** panel, issue the ALT line command against the foreign key that you want to change.



```
ADB21TFK  -- DB2X Foreign Keys of Table DSN91010.DE >  ------------ Row 1 of 2
Command ===>                                                  Scroll ===> PAGE

Line commands:
 FC - From Column   TC - To Column   T - To Table    ALT - Alter FK
 ? - Show all line commands
    From:                       To:
Sel Column Name        Rel Name Schema   Name              Column Name
    *                  *        *        *                 *
--- ------------------ -------- -------- ----------------- -------------------
alt ADMRDEPT           RDD      DSN91010 DEPT              DEPTNO
    MGRNO              RDE      DSN91010 EMP               EMPNO
******************************* END OF DB2 DATA *******************************
```

*Figure 269. **Foreign Keys of Table (ADB21TFK)** panel - Changing a foreign key*

5. On the **Alter Foreign Key Constraint (ADB21TAF)** panel, make changes to the foreign key attributes.



```
 ADB21TAF  -------------- DB2X Alter Foreign Key Constraint -------------- 08:20
Command ===>

Commands: COLUMNS

                                                                   More:     +
 ALTER TABLE

 Table schema . . .  DSN91010   >
 Table name . . . .  DEPT              >

 FOREIGN KEY
 Constraint name  . . RDD01              > (? to look up existing constraints

 Columns
        (  ADMRDEPT,MGRNO



                                                                            > )

 REFERENCES    Table schema . . . DSN91010  >
               Table name . . . . DEPT               > (? to look up

 ON DELETE . . . .           (RESTRICT, CASCADE, SET NULL, or NO ACTION)
```

Figure 270. Alter Table (ADB21TAF)

6. Press Enter.

   The **Alter Objects (ADB27CA)** panel is displayed.
7. Generate an ALT job

Related concepts

“ALT (Alter) function” on page 447

When you use the ALT line command to change an object, you invoke the ALT function of Db2 Admin Tool. The ALT function first analyzes the change to determine what actions need to be taken to implement the change. You can choose whether you want this analysis to be done in batch or online.

Altering sequence aliases

You can alter a sequence alias to specify a different *target sequence*, or sequence on which the alias is defined.

About this task

Restriction: The target sequence cannot be any of the following items:

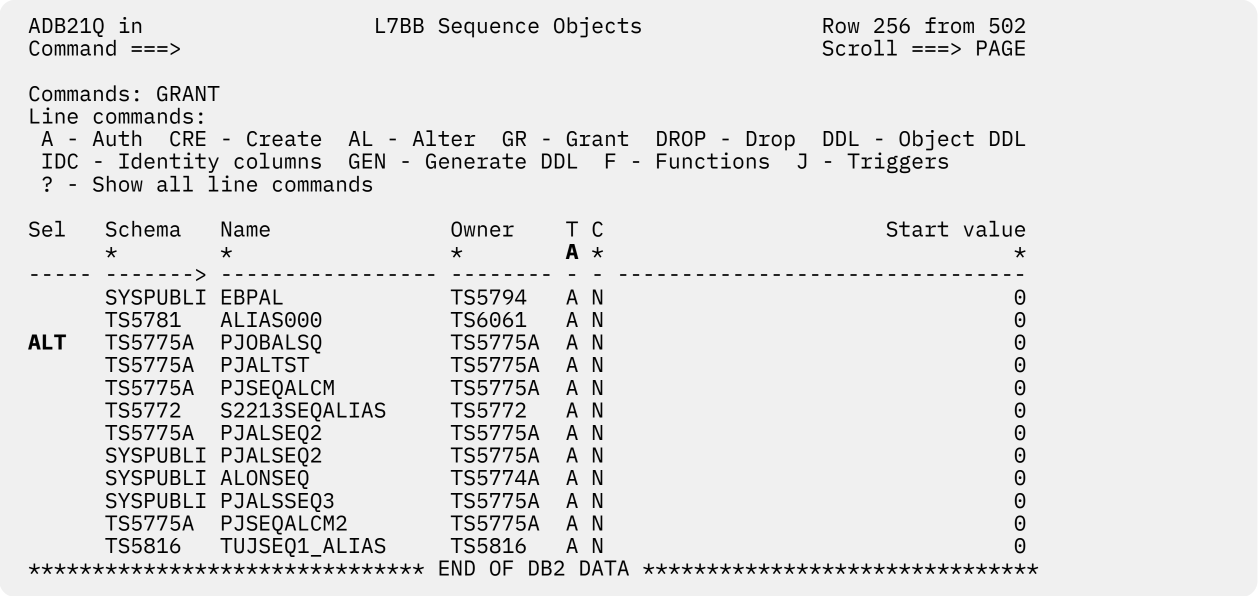
- An existing sequence alias
- A sequence generated for an identity column
- A sequence generated for a DOCID column

Procedure

To alter a sequence alias:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify any selection criteria at the bottom, specify option Q, and press Enter.
3. On the **Sequence Objects (ADB21Q)** panel, specify ALT against the alias that you want to alter, and press Enter.

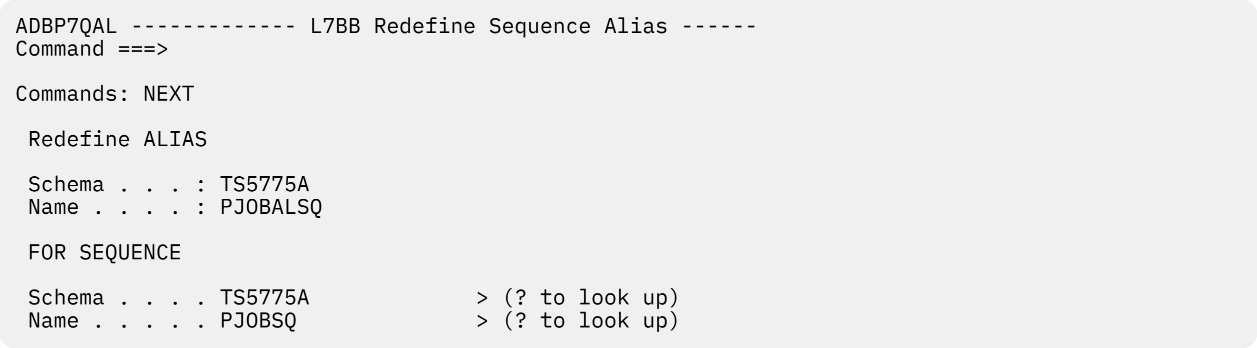
   Tip: To filter the list to display only aliases, type A in the search argument area in the **T** column, and press Enter.



```
ADB21Q in                  L7BB Sequence Objects              Row 256 from 502
Command ===>                                                  Scroll ===> PAGE

Commands: GRANT
Line commands:
 A - Auth  CRE - Create  AL - Alter  GR - Grant  DROP - Drop  DDL - Object DDL
 IDC - Identity columns  GEN - Generate DDL  F - Functions  J - Triggers
 ? - Show all line commands

Sel   Schema   Name              Owner    T C                        Start value
      *        *                 *        A *                                  *
----- -------> ----------------- -------- - - -------------------------------
      SYSPUBLI EBPAL             TS5794   A N                                  0
      TS5781   ALIAS000          TS6061   A N                                  0
ALT   TS5775A  PJOBALSQ          TS5775A  A N                                  0
      TS5775A  PJALTST           TS5775A  A N                                  0
      TS5775A  PJSEQALCM         TS5775A  A N                                  0
      TS5772   S2213SEQALIAS     TS5772   A N                                  0
      TS5775A  PJALSEQ2          TS5775A  A N                                  0
      SYSPUBLI PJALSEQ2          TS5775A  A N                                  0
      SYSPUBLI ALONSEQ           TS5774A  A N                                  0
      SYSPUBLI PJALSSEQ3         TS5775A  A N                                  0
      TS5775A  PJSEQALCM2        TS5775A  A N                                  0
      TS5816   TUJSEQ1_ALIAS     TS5816   A N                                  0
******************************* END OF DB2 DATA *******************************
```

*Figure 271. **Sequence Objects (ADB21Q)** panel*

4. On the **Redefine Sequence Alias (ADBP7QAL)** panel, specify a different target sequence under **FOR SEQUENCE**:

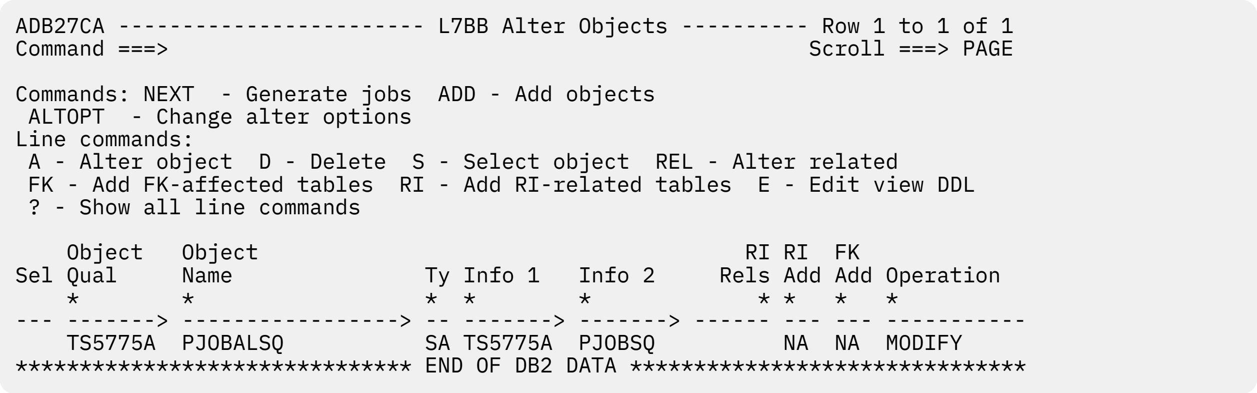


```
ADBP7QAL ------------- L7BB Redefine Sequence Alias ------
Command ===>

Commands: NEXT

 Redefine ALIAS

 Schema . . . : TS5775A
 Name . . . . : PJOBALSQ

 FOR SEQUENCE

 Schema . . . . TS5775A             > (? to look up)
 Name . . . . . PJOBSQ              > (? to look up)
```

Figure 272. ***Redefine Sequence Alias (ADBP7QAL)*** *panel*

5. Specify NEXT, and press Enter.

   The **Alter Objects (ADB27CA)** panel is displayed. Notice that the **Ty** column for your object indicates the type is SA for sequence alias.



```
ADB27CA ------------------------ L7BB Alter Objects ---------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Commands: NEXT  - Generate jobs  ADD - Add objects
 ALTOPT  - Change alter options
Line commands:
 A - Alter object  D - Delete  S - Select object  REL - Alter related
 FK - Add FK-affected tables  RI - Add RI-related tables  E - Edit view DDL
 ? - Show all line commands

    Object   Object                                      RI RI  FK
Sel Qual     Name               Ty Info 1   Info 2     Rels Add Add Operation
    *        *                  *  *        *             * *   *   *
--- -------> -----------------> -- -------> -------> ------ --- --- -----------
    TS5775A  PJOBALSQ           SA TS5775A  PJOBSQ          NA  NA  MODIFY
******************************* END OF DB2 DATA *******************************
```

6. Generate an ALT job.

Generating an ALT job

When using the ALT command, you specify the changes that you want to make to the Db2 object or objects. Then you generate a job to implement those changes. The process of generating this job is the same for all object types.

Before you begin

You must have already completed one of the following tasks and have the **Alter Objects (ADB27CA)** panel displayed:

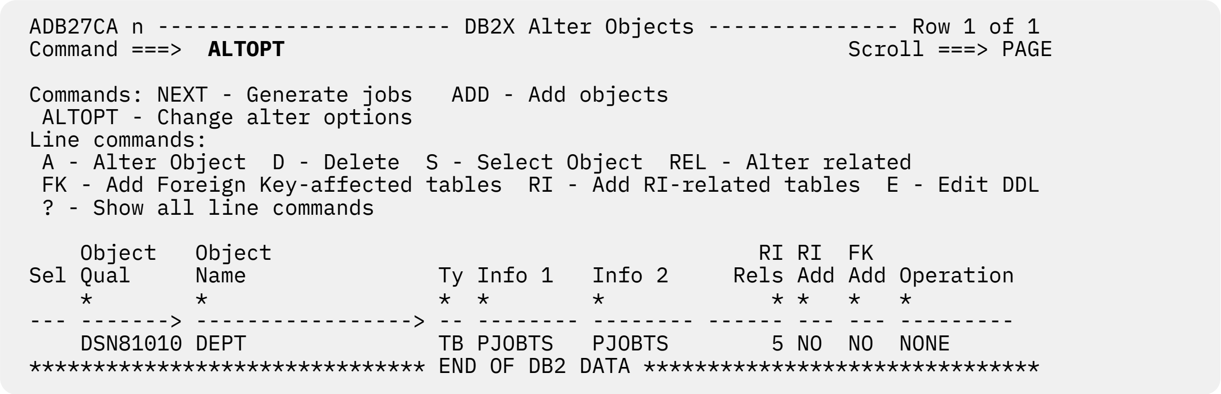
- “Altering a database by using the ALT line command” on page 451
- “Altering a table space by using the ALT command” on page 454
- “Redefining partitions in a partitioned table space that uses index-controlled partitioning” on page 456
- “Redefining partitions in a partitioned table space that uses table-controlled partitioning” on page 459
- “Altering a table by using the ALT line command” on page 461
- “Changing the related objects for a table” on page 481
- “Altering a partition in an archive table” on page 482
- “Adding LOB columns to an existing table” on page 485
- “Altering an index by using the ALT line command” on page 491
- “Renaming indexes” on page 493
- “Excluding null keys from an index” on page 494
- “Altering views” on page 497

- “Altering sequence aliases” on page 499

Procedure

To generate an ALT job:

1. Optional: If you want to change the method that Db2 Admin Tool uses (batch or online) to analyze the alter operation, complete the following steps:
   a) On the **Alter Objects (ADB27CA)** panel, issue the ALTOPT command, and press Enter:



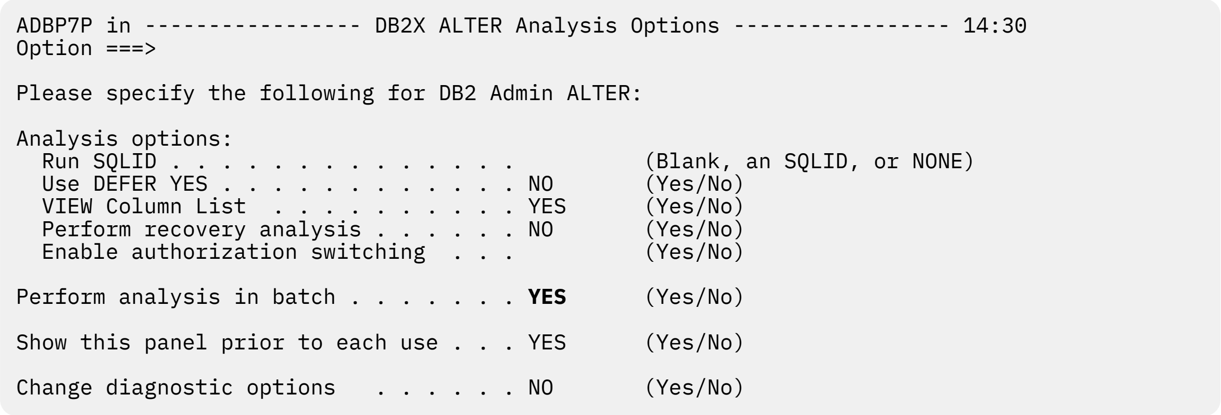
```
ADB27CA n ----------------------- DB2X Alter Objects --------------- Row 1 of 1
Command ===>  ALTOPT                                             Scroll ===> PAGE

Commands: NEXT - Generate jobs   ADD - Add objects
 ALTOPT - Change alter options
Line commands:
 A - Alter Object  D - Delete  S - Select Object  REL - Alter related
 FK - Add Foreign Key-affected tables  RI - Add RI-related tables  E - Edit DDL
 ? - Show all line commands

    Object   Object                              RI RI  FK
Sel Qual     Name               Ty Info 1   Info 2     Rels Add Add Operation
    *        *                  *  *        *            * *   *   *
--- -------> -----------------> -- -------- -------- ------ --- --- ---------
    DSN81010 DEPT               TB PJOBTS   PJOBTS        5 NO  NO  NONE
******************************* END OF DB2 DATA *******************************
```

*Figure 273. **Alter Objects (ADB27CA)** panel*

   b) On the **ALTER Options Menu (ADBP7OP)** panel, select option 1, and press Enter.
   c) On the **ALTER Analysis Options (ADBP7P)** panel, specify one of the following values in the **Perform analysis in batch** field and press Enter:

   YES
      Specifies batch analysis, which is the preferred method.

   NO
      Specifies online analysis.



```
ADBP7P in ----------------- DB2X ALTER Analysis Options ----------------- 14:30
Option ===>

Please specify the following for DB2 Admin ALTER:

Analysis options:
  Run SQLID . . . . . . . . . . . . . .          (Blank, an SQLID, or NONE)
  Use DEFER YES . . . . . . . . . . . . NO       (Yes/No)
  VIEW Column List  . . . . . . . . . . YES      (Yes/No)
  Perform recovery analysis . . . . . . NO       (Yes/No)
  Enable authorization switching  . . .          (Yes/No)

Perform analysis in batch . . . . . . . YES      (Yes/No)

Show this panel prior to each use . . . YES      (Yes/No)

Change diagnostic options   . . . . . . NO       (Yes/No)
```

*Figure 274. **ALTER Analysis Options (ADBP7P)** panel*

   d) Exit back to the **Alter Objects (ADB27CA)** panel.
2. On the **Alter Objects (ADB27CA)** panel, issue the NEXT command, and press Enter.

   If online analysis was specified (**Perform analysis in batch** = NO), one of the following actions occurs:

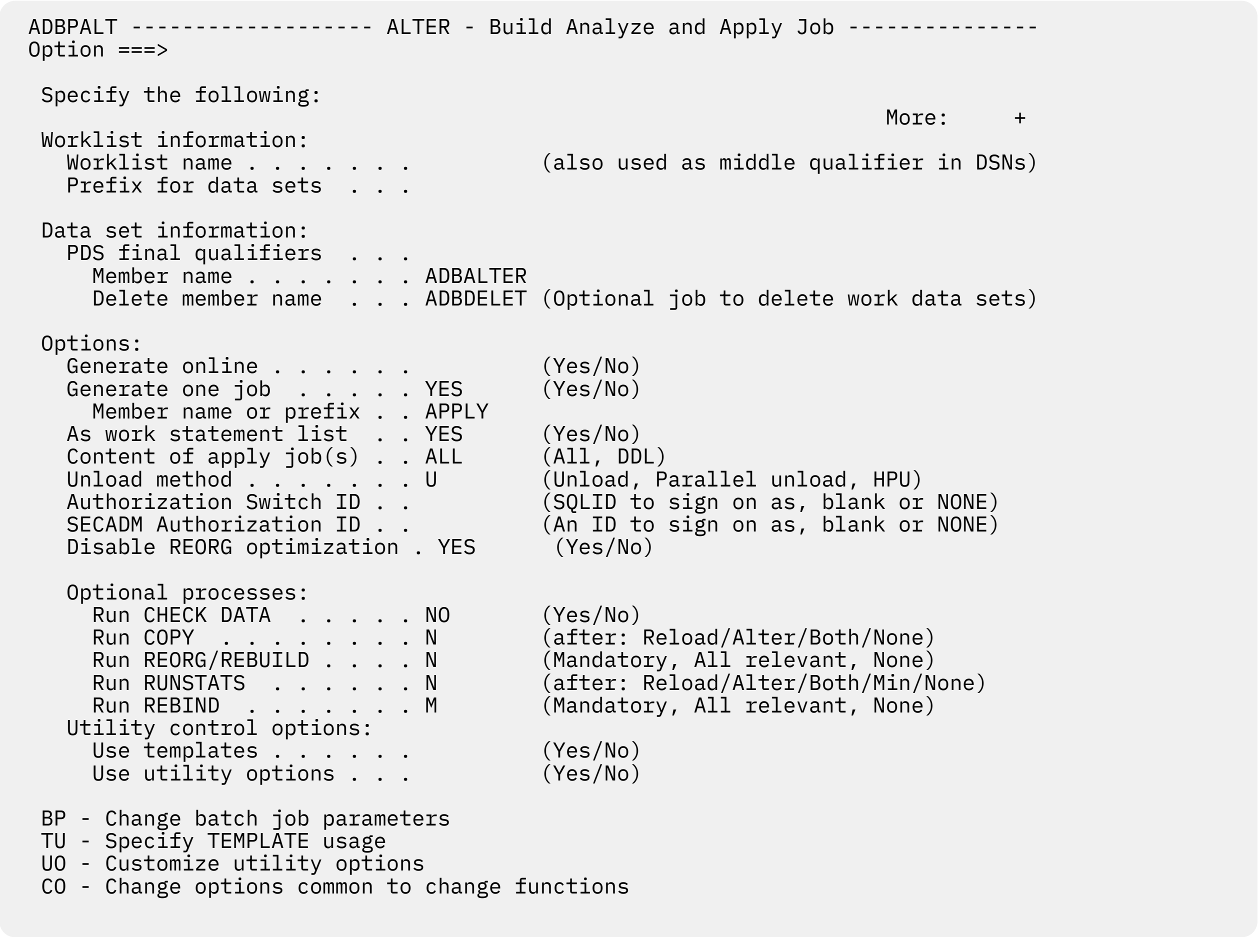
   - If the analysis process determines that SQL ALTER statements can accomplish the task, **ADB27CTC** panel is displayed. On this panel, specify whether you want to run the SQL statements in the foreground (online) or generate a batch job.

     If you select ALTER statements, the SQL is run in the foreground. In this case, when the ALTER statement completes successfully, the change is made, and you can skip the remaining steps.

If you specify batch jobs, the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed.

- If the analysis process determines that an SQL ALTER statement cannot be used, the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed.

3. On the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel, specify options for building the WSL or batch job to implement the changes, and press Enter:



```
ADBPALT ------------------- ALTER - Build Analyze and Apply Job ---------------
Option ===>

 Specify the following:
                                                                   More:     +
 Worklist information:
   Worklist name . . . . . . .          (also used as middle qualifier in DSNs)
   Prefix for data sets  . . .

 Data set information:
   PDS final qualifiers  . . .
     Member name . . . . . . . ADBALTER
     Delete member name  . . . ADBDELET (Optional job to delete work data sets)

 Options:
   Generate online . . . . . .          (Yes/No)
   Generate one job  . . . . . YES      (Yes/No)
     Member name or prefix . . APPLY
   As work statement list  . . YES      (Yes/No)
   Content of apply job(s) . . ALL      (All, DDL)
   Unload method . . . . . . . U        (Unload, Parallel unload, HPU)
   Authorization Switch ID . .          (SQLID to sign on as, blank or NONE)
   SECADM Authorization ID . .          (An ID to sign on as, blank or NONE)
   Disable REORG optimization . YES      (Yes/No)

   Optional processes:
     Run CHECK DATA  . . . . . NO       (Yes/No)
     Run COPY  . . . . . . . . N        (after: Reload/Alter/Both/None)
     Run REORG/REBUILD . . . . N        (Mandatory, All relevant, None)
     Run RUNSTATS  . . . . . . N        (after: Reload/Alter/Both/Min/None)
     Run REBIND  . . . . . . . M        (Mandatory, All relevant, None)
   Utility control options:
     Use templates . . . . . .          (Yes/No)
     Use utility options . . .          (Yes/No)

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

*Figure 275. **ALTER - Build Analyze and Apply Job (ADBPALT)** panel*

4. If any confirmation panels are displayed, confirm or change the data set name or WSL, and press Enter.
5. Submit the generated job to analyze the change and build the apply jobs or WSL for the change.

   For more information about the jobs that are generated, see "ALT (Alter) function" on page 447. For more information about the data sets that are used by ALT, see "Work data sets used by the ALT (Alter Redefine) function" on page 449.

Transferring ownership of objects from one user or role to another

When business responsibilities change, it is often necessary to change the ownership of objects from one user or role to another.

Procedure

To transfer ownership of objects from one user or role to another:

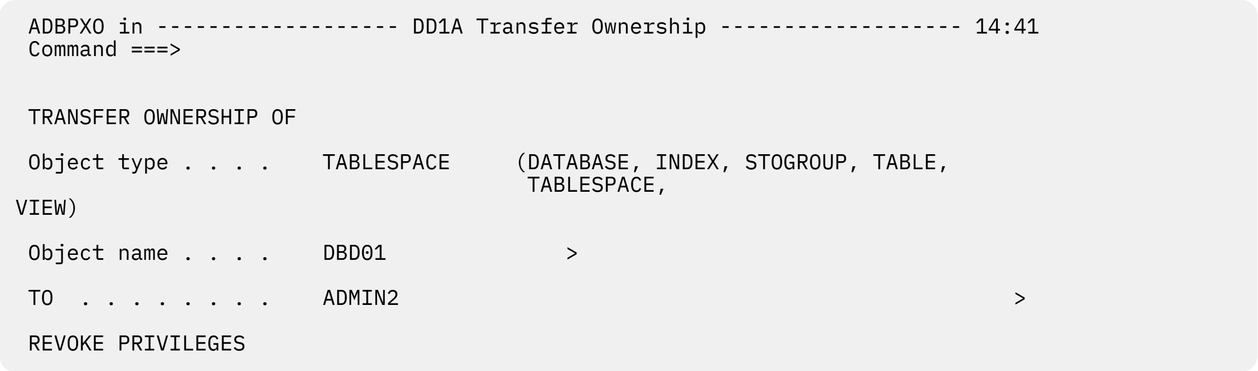
1. On the panel that corresponds to the type of object for which you want to transfer ownership, find the row for the object, specify the line command XO in the **Select** column, and press Enter.

   You can transfer ownership of the objects on the following panels:

   Aliases (ADB21A)

Databases (ADB21D)
Grant/Revoke Privileges On Objects (ADB2G)
Indexes (ADB21X)
Storage Groups (ADB21G)
Table Spaces (ADB21S)
Tables, Views, and Aliases (ADB21T)
Views (ADB21VV) panel

The **Transfer Ownership** panel is displayed, as shown in the following figure:



```
 ADBPXO in ------------------- DD1A Transfer Ownership ------------------- 14:41
 Command ===>


 TRANSFER OWNERSHIP OF

 Object type . . . .    TABLESPACE     (DATABASE, INDEX, STOGROUP, TABLE,
                                        TABLESPACE,
VIEW)

 Object name . . . .    DBD01              >

 TO  . . . . . . . .    ADMIN2                                              >

 REVOKE PRIVILEGES
```

2. In the **TO** field, specify the ID of the new owner according to the following considerations:
   - If the new owner is an authorization ID, specify the authorization ID.
   - If the new owner is a role, specify ROLE *role-name*.
   - If the object is a session variable, specify SESSION_USER.

Moving tables from multi-table table spaces to UTS

You can use Db2 Admin Tool to move tables from multi-table simple or segmented table spaces, which are deprecated, to partition-by-growth universal table spaces (UTS). Moving data to supported table spaces can help you take advantage of new Db2 functionality that operates only on UTS.

Before you begin

To perform this task, you must be running Db2 12 for z/OS function level 508 or higher.

About this task

For more detailed information about moving tables from multi-table table spaces in Db2, see Moving tables from multi-table table spaces to partition-by-growth table spaces (Db2 12 for z/OS) .

Procedure

To move tables from multi-table table spaces to UTS:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option **1**, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify one of the following options and any filtering criteria at the bottom of the panel, and press Enter:

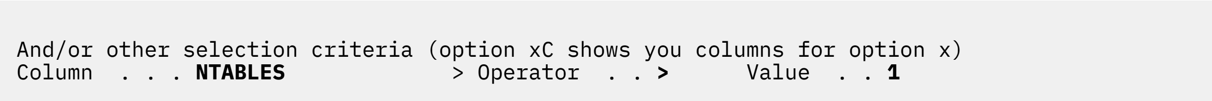
   S

   Specify this option if you want to select the tables to move based on table spaces.

   D

   Specify this option if you want to select the tables to move based on databases.

   Tip: If you specify S, also specify the following filtering criteria to limit the results to only those table spaces that contain more than one table:

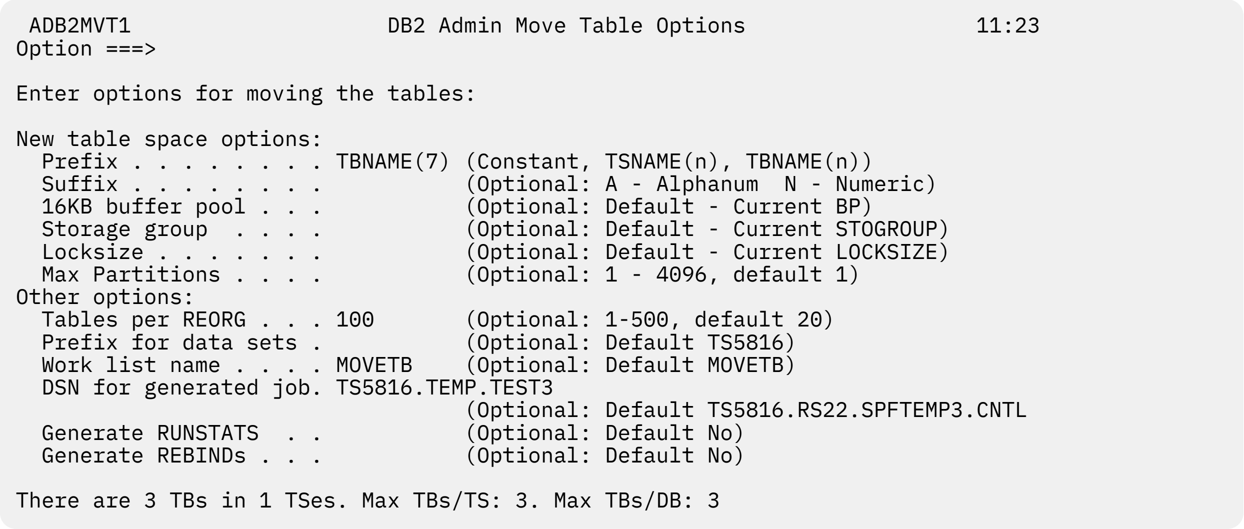


```
And/or other selection criteria (option xC shows you columns for option x)
Column  . . . NTABLES             > Operator  . . >      Value  . . 1
```

3. On the **Table Spaces (ADB21S)** panel or the **Databases (ADB21D)** panel, specify either the MOVETB line command or primary command.

   Use the line command to select a specific table space or database. Use the primary command to select all objects listed. When you specify the primary command, any eligible multi-table table spaces are selected.

4. On the **DB2 Admin Move Table Options (ADB2MVT1)** panel, specify options for the move operation and press Enter:

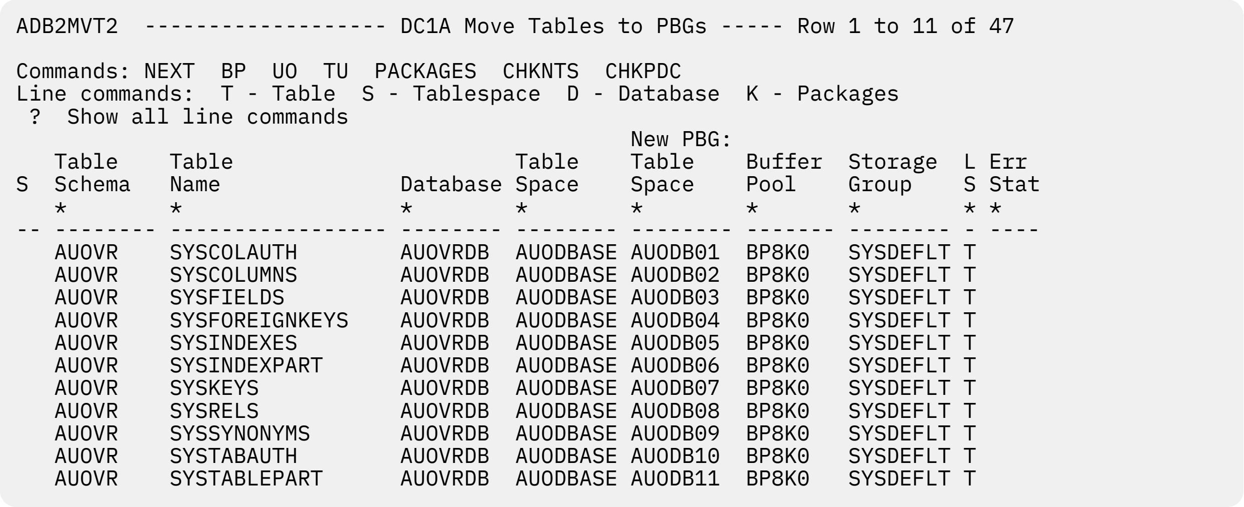


```
 ADB2MVT1                    DB2 Admin Move Table Options                  11:23
Option ===>

Enter options for moving the tables:

New table space options:
  Prefix . . . . . . . . TBNAME(7) (Constant, TSNAME(n), TBNAME(n))
  Suffix . . . . . . . .           (Optional: A - Alphanum  N - Numeric)
  16KB buffer pool . . .           (Optional: Default - Current BP)
  Storage group  . . . .           (Optional: Default - Current STOGROUP)
  Locksize . . . . . . .           (Optional: Default - Current LOCKSIZE)
  Max Partitions . . . .           (Optional: 1 - 4096, default 1)
Other options:
  Tables per REORG . . . 100       (Optional: 1-500, default 20)
  Prefix for data sets .           (Optional: Default TS5816)
  Work list name . . . . MOVETB    (Optional: Default MOVETB)
  DSN for generated job. TS5816.TEMP.TEST3
                                   (Optional: Default TS5816.RS22.SPFTEMP3.CNTL
  Generate RUNSTATS  . .           (Optional: Default No)
  Generate REBINDs . . .           (Optional: Default No)

There are 3 TBs in 1 TSes. Max TBs/TS: 3. Max TBs/DB: 3
```

   You must specify a prefix for the names of the new table spaces to which the tables are to be moved. You can either specify a constant value or a subset of characters from the database, table space, or table name. All other values are optional and are described in the help.

   Tip: REORG operations are used to materialize pending MOVE TABLE operations. When you specify the number of tables to include in a REORG, consider that the number of tables and size of these tables can affect the elapsed time of REORG and the storage that REORG requires for work files.

5. On the **Move Tables to PBGs (ADB2MVT2)** panel, confirm that the tables that you want to move are listed:



```
ADB2MVT2  ------------------- DC1A Move Tables to PBGs ----- Row 1 to 11 of 47

Commands: NEXT  BP  UO  TU  PACKAGES  CHKNTS  CHKPDC
Line commands:  T - Table  S - Tablespace  D - Database  K - Packages
 ?  Show all line commands
                                                New PBG:
   Table    Table                     Table    Table    Buffer  Storage  L Err
S  Schema   Name             Database Space    Space    Pool    Group    S Stat
   *        *                *        *        *        *       *        * *
-- -------- ---------------- -------- -------- -------- ------- -------- - ----
   AUOVR    SYSCOLAUTH       AUOVRDB  AUODBASE AUODB01  BP8K0   SYSDEFLT T
   AUOVR    SYSCOLUMNS       AUOVRDB  AUODBASE AUODB02  BP8K0   SYSDEFLT T
   AUOVR    SYSFIELDS        AUOVRDB  AUODBASE AUODB03  BP8K0   SYSDEFLT T
   AUOVR    SYSFOREIGNKEYS   AUOVRDB  AUODBASE AUODB04  BP8K0   SYSDEFLT T
   AUOVR    SYSINDEXES       AUOVRDB  AUODBASE AUODB05  BP8K0   SYSDEFLT T
   AUOVR    SYSINDEXPART     AUOVRDB  AUODBASE AUODB06  BP8K0   SYSDEFLT T
   AUOVR    SYSKEYS          AUOVRDB  AUODBASE AUODB07  BP8K0   SYSDEFLT T
   AUOVR    SYSRELS          AUOVRDB  AUODBASE AUODB08  BP8K0   SYSDEFLT T
   AUOVR    SYSSYNONYMS      AUOVRDB  AUODBASE AUODB09  BP8K0   SYSDEFLT T
   AUOVR    SYSTABAUTH       AUOVRDB  AUODBASE AUODB10  BP8K0   SYSDEFLT T
   AUOVR    SYSTABLEPART     AUOVRDB  AUODBASE AUODB11  BP8K0   SYSDEFLT T
```

6. To check whether any of the new table spaces already exist, issue the CHKNTS command, and press Enter.

If any of the new table spaces already exist, a list of those table space names is displayed. These existing table spaces will cause the generated move job to fail, because that job will attempt to create these table spaces. Therefore, in the generated job, the operations for these table spaces will be placed in comments and the tables will not be moved. To resolve this issue and ensure that the tables are moved, exit back to the **Move Tables to PBGs (ADB2MVT2)** panel and specify a new name in the **New PBG: Table Space** column for those table spaces that already exist. Then, issue CHKNTS again to clear the E status in the **Err Stat** column.

The **Err Stat** column can have the following values:

D
The new table space name is a duplicate of a new table space in the list. To resolve this error, enter a different table space name.

P
The old table space already has a pending change that prevents additional ALTER operations. To resolve this error, first materialize or delete the existing pending change.

E
The new table space already exists in the Db2 catalog. To resolve this error, enter a different table space name.

7. To check whether any of the existing table spaces have pending changes, issue the CHKPDC command, and press Enter.

   If any pending changes exist, a list of the affected table spaces is displayed. Those pending changes will also cause the generated move job to fail, because Db2 does not allow you to move tables from a table space with pending changes. Therefore, in the generated job, the operations for these table spaces will be placed in comments and the tables will not be moved. To resolve this issue and ensure that the tables are moved, exit back to the **Move Tables to PBGs (ADB2MVT2)** panel and take the following actions for each table space with a pending change:

   a. Specify the S line command next to any row that includes the table space with pending changes.
   b. On the **Table Spaces (ADB21S)** panel, specify the PDC line command to view the pending changes.
   c. On the **DB2 Pending Definition Changes (ADBPPDC)** panel, evaluate the changes and either drop them or run REORG on the table space.

   Then, issue CHKPDC again to clear the P status in the **Err Stat** column.
8. Optional: Modify the REORG options by using the UO command.
9. Issue the NEXT command, and press Enter.

   The generated JCL job is displayed. This job performs the following actions:

   - Creates the new tables spaces
   - Runs the ALTER TABLESPACE statements with the MOVE TABLE clause and the required REORG operations to materialize these changes
   - Performs any requested rebind operations

   The number of tables that are moved and the number of table moves that are materialized in each REORG are listed at the top of the job, before the CREATE TABLESPACE statements.
10. Submit the generated JCL to move the tables.

    Because this job runs ADBTEP2, if it fails, you can resubmit it and ADBTEP2 will restart it.

Related information

Video: Db2 Administration Tool: moving tables from multi-table table spaces into universal table spaces (FL508) (IBM community: Db2 Tools for z/OS)

Invoking Db2 Table Editor from Db2 Admin Tool

IBM Db2 Table Editor for z/OS is a product that makes it easy to view and modify your Db2 table data. If you have Db2 Table Editor installed, you can launch it from Db2 Admin Tool.

Before you begin

To be able to invoke Db2 Table Editor from Db2 Admin Tool, Db2 Table Editor must be enabled during the customization of Db2 Admin Tool. Specifically, the following fields on the **DB2 Parameters (CCQPDB2)** panel must be set:

- **Enable Db2 Table Editor** must be set to YES
- **Table Editor CLIST(mbr)** must list the CLIST member

Optionally, set the **Pass accelerator name to Table Editor** field to indicate whether you want accelerator information to be passed to Db2 Table Editor.

For details on setting these values during customization, see "Defining Db2 parameters" on page 106.

Procedure

To invoke Db2 Table Editor from Db2 Admin Tool:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option T and optionally any search criteria at the bottom of the panel to find the table that you want to edit, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, specify the EDIT line command next to the desired table, and press Enter.

   If **Pass accelerator name to Table Editor** was set to YES on the **DB2 Parameters (CCQPDB2)** panel and the table to be edited is associated with at least one accelerator, the **DB2 Accelerators (ADBPZAC)** panel displays a list of accelerators that are related to the selected table. Otherwise, Db2 Table Editor opens, and no accelerators are listed.
4. If the **DB2 Accelerators (ADBPZAC)** panel is displayed, use the following commands to select the accelerators that you want Db2 Table Editor to use, and press Enter:

   +

   Specify this line command next to any accelerator to select it.

   ALL

   Specify this primary command on the command line to select all accelerators listed.

   NONE

   Specify this primary command on the command line to specify that you do not want to pass any accelerator information to Db2 Table Editor.

   The selected accelerator or accelerators must be active. If a selected accelerator is not active, the accelerator name is not passed to Db2 Table Editor. If none of the selected accelerators are active, the following message is displayed to prompt you to activate at least one of them:

   ```
   Selected acceletor is not active, please start it or select command NONE to work without
   accelerator
   ```

What to do next

For instructions on how to use Db2 Table Editor, see Db2 Table Editor 4.5.0.

Details about objects in batch mode

You don't have to use the ISPF panels to generate the reports that contain details about tables, packages, and accelerated tables. Retrieving the details in batch mode can improve deployment and adoption processes.

The results that the detailed batch report generates in the REPORT DD of the job output are similar to the results that are generated by the online DET function.

The following figure shows an example of a generated batch job:

```
//GETDT1 JOB (ACCOUNTING-INFO),'DB2 UTILITY',
//*       RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
//         REGION=0M,NOTIFY=TS5771,
//         MSGCLASS=H,
//         CLASS=A
//*
//ADBLIBS JCLLIB ORDER=ADB.DEVCUST.PROCLIB
//*
//*
//*******************************************************ADB2SPFB***
//* DB2 ADMIN ISPF BATCH
//******************************************************************

//S01DET  EXEC PGM=IKJEFT01,DYNAMNBR=100

//SYSEXEC  DD DISP=SHR,DSN=DMTOOL.SADBEXEC

//         DD DISP=SHR,DSN=ADB.VC1APAR.EXEC

//         DD DISP=SHR,DSN=GOC.VC1APAR.EXEC

//SYSPROC  DD DISP=SHR,DSN=DMTOOL.SADBCLST

//         DD DISP=SHR,DSN=ADB.VC1APAR.CLIST

//         DD DISP=SHR,DSN=GOC.VC1APAR.CLIST

//ISPLLIB  DD DISP=SHR,DSN=DMTOOL.SADBLLIB

//         DD DISP=SHR,DSN=ADB.VC1APAR.ISPLLIB

//         DD DISP=SHR,DSN=GOC.VC1APAR.ISPLLIB

//STEPLIB  DD DISP=SHR,DSN=DMTOOL.SADBLLIB

//         DD DISP=SHR,DSN=ADB.VC1APAR.ISPLLIB

//         DD DISP=SHR,DSN=GOC.VC1APAR.ISPLLIB

//         DD DISP=SHR,DSN=DC1A.SDSNEXIT

//         DD DISP=SHR,DSN=DSN.VC10.SDSNLOAD

//         DD DISP=SHR,DSN=DSN.VC10.SDSNLOD2

//         DD DISP=SHR,DSN='ISP.SISPLOAD'

//ISPMLIB  DD DISP=SHR,DSN=DMTOOL.SADBMLIB

//         DD DISP=SHR,DSN=ADB.VC1APAR.ISPMLIB

//         DD DISP=SHR,DSN=GOC.VC1APAR.ISPMLIB

//         DD DISP=SHR,DSN=ISP.SISPMENU

//ISPPLIB  DD DISP=SHR,DSN=DMTOOL.SADBPLIB

//         DD DISP=SHR,DSN=ADB.VC1APAR.ISPPLIB

//         DD DISP=SHR,DSN=GOC.VC1APAR.ISPPLIB

//ISPSLIB  DD DISP=SHR,DSN=DMTOOL.SADBSLIB

//         DD DISP=SHR,DSN=ADB.VC1APAR.ISPSLIB

//         DD DISP=SHR,DSN=GOC.VC1APAR.ISPSLIB
```

Figure 276. Example of a generated batch job (Part 1)

```
//ISPTLIB  DD DSN=&ISPTLIB,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=80,DSORG=PO),
//            SPACE=(80,(1,5,10)),UNIT=SYSALLDA,AVGREC=K
//         DD DISP=SHR,DSN=DMTOOL.SADBTLIB
//         DD DISP=SHR,DSN=ADB.VC1APAR.ISPTLIB
//         DD DISP=SHR,DSN=GOC.VC1APAR.ISPTLIB
//         DD DISP=SHR,DSN=ISP.SISPTENU
//ISPPROF  DD DSN=&ISPPROF,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=7920,DSORG=PO)
//            SPACE=(80,(1,5,10)),UNIT=SYSALLDA,AVGREC=K
//ISPFILE  DD DSN=&ISPFILE,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=7920,DSORG=PO)
//            SPACE=(80,(1,10,10)),UNIT=SYSALLDA,AVGREC=K
//ISPCTL0  DD DSN=&ISPCTL0,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=7920,DSORG=PS)
//            SPACE=(80,(0,5)),UNIT=SYSALLDA,AVGREC=K
//ISPCTL1  DD DSN=&ISPCTL1,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=7920,DSORG=PS)
//            SPACE=(80,(0,5)),UNIT=SYSALLDA,AVGREC=K
//ISPCTL2  DD DSN=&ISPCTL2,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=7920,DSORG=PS)
//            SPACE=(80,(0,5)),UNIT=SYSALLDA,AVGREC=K
//ISPWRK1  DD DSN=&ISPWRK1,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=256,BLKSIZE=256,DSORG=PS)
//            SPACE=(CYL,(5,10)),UNIT=SYSALLDA
//ISPWRK2  DD DSN=&ISPWRK2,DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=256,BLKSIZE=256,DSORG=PS)
//            SPACE=(CYL,(5,10)),UNIT=SYSALLDA
//ISPLOG   DD SYSOUT=*,DCB=(LRECL=125,BLKSIZE=129,RECFM=VA)
//ISPLIST  DD SYSOUT=*,DCB=(LRECL=125,BLKSIZE=129,RECFM=VA)
```

Figure 277. Example of a generated batch job (Part 2)

```
//REPORT   DD SYSOUT=*

//ADBDIAG  DD SYSOUT=*

//SYSPRINT DD SYSOUT=*

//SYSTSPRT DD SYSOUT=*

//SYSTSIN  DD *

ISPSTART CMD(%ADBRPGM TS5771P ADBDET DB2SYS(DC1A))

//**********************************************************************
//* END OF ISPF BATCH STEP SET UP
//**********************************************************************
//*
//*************************************************************ADBSBET**
//* START OF PARMS SETUP
//**********************************************************************
//PARMS DD *

TYPE = 'TB' QUAL = 'TS5771' NAME = 'RHPTB%';
TYPE = 'PK' QUAL = 'TDKDB' NAME = 'TRDEL' VERSION = '*';
TYPE = 'AT' QUAL = 'TS577%' NAME = 'T%' ACCELERATOR = '%ACC1';
```

Figure 278. Example of a generated batch job (Part 3)

Depending on the object type, you might include the following parameters.

- TYPE
- QUAL
- NAME
- VERSION
- ACCELERATOR

Before you specify values for these parameters, consider the following points that apply to all object types:

- Each TYPE record entry must terminate with a semi-colon (;) to indicate the end of its input parameters.
- The wild cards % or * (all) can be specified for the QUAL, NAME, VERSION and ACCELERATOR parameters.
- The TYPE parameter is required. The other parameters are optional. For example, leaving the QUAL parameter blank is equivalent to specifying QUAL=`'*'`.

Before you specify values for these parameters, consider the following points that apply to only certain object types:

Tables

- The BET command filters and generates batch jobs only for the following types of tables: C, G, H, P, R, T, X.
- When wild cards are used, details will be generated for above valid table types only. All other table types, which are pulled by using wild cards, are skipped and displayed in SYSPRINT DD. The job will end with RC=4.

Packages

- Each package can include many SQL statements. To avoid long delays or storage and memory issues, consider using absolute values instead of wild cards.

Accelerated tables

- The BET command filters and generates batch jobs only for tables that exist in the SYSIBM.SYSTABLES Db2 catalog table.
- If an error occurs, such as when the definition of a table on Db2 and the accelerator are out of sync, in any table in the list, the error details are displayed in REPORT DD along with details about other

valid tables, and the job ends with RC=4. If no other error-free, valid table details can be displayed, the job ends with RC=8. The SYSPRINT DD lists tables that were in error.

The following figure shows an example of a report:

```
======================

Details for Objects Report

======================



Tables  . . . . . . . . . . . . . : 1

Packages  . . . . . . . . . . . . : 1

Partitioned Accelerated Tables  . : 1

Non-Partitioned Accelerated Tables: 1



Total number of objects . . . . . : 4



=========

Object #1

=========



 Details for table (label) : TS5771.RHPTB1

  Table information
    Table schema . . . : TS5771        Table name . . . . : RHPTB1
    Created by . . . . : TS5771        Created  . : 2017-06-27-13.49.43.922171
    Table space name . : RHPTS1        Database name  . . : RHPDB1
    Object ID for table: 3             DB ID for database : 549
    Maximum row length : 100           Primary key OBID . : N/A
    Number of columns  : 9             Primary key columns: N/A
    Validate procedure : N/A           EDIT procedure name: N/A
                                       With row attributes: N/A
    Parent relations . : 0             Child relations  . : 0
    Auditing . . . . . : AUDIT NONE    Status . . : No primary key
    Data capture . . . : NO            Altered  . : 2017-06-27-13.49.43.922171
    Restrict on DROP . : NO            Check constraints  : None
    Encoding scheme  . : U - UNICODE   Col. in part. key  : 0
    Check flag . . . . : No              VOLATILE table . . : No
    Created in DB2 Ver : Q             Dependent MQTS . . : 0
    Data version . . . : 0
    Table owner  . . . : TS5771
    Owner type . . . . : Auth ID       Append specified . : No
    Clone table schema :               Clone table name . :
    Access control . . : ' ' - Not enforced
    Number of hash cols: 0
    Versioning schema  :               Versioning table . :
    Archiving schema . :               Archiving table  . :
    Table creation RBA : 00000000008B97D148BA (Hex)
    Last alter RBA . . : 00000000008B97D148BA (Hex)

    Statistical data . : No valid data available
    Stats feedback . . : Yes - statistics recommendations are collected
```

Figure 279. Example of a report (Part 1)

```
     Associated remarks :

   Column information for table : TS5771.RHPTB1

     Column Name        Col No Col Type Length  Scale Null Def FP    Col card
     ------------------ ------ -------- ------ ------ ---- --- -- -----------
     EMPNO                   1 CHAR          6      0 N    N   N           -1
     EMPNAME                 2 VARCHAR      30      0 Y    Y   N           -1
     HIREDATE                3 DATE          4      0 Y    Y   N           -1
     LB1                     4 CLOB          4      0 Y    Y   N           -1
     LB2                     5 CLOB          4      0 Y    Y   N           -1
     ISMANAGER               6 INTEGER       4      0 Y    Y   N           -1
     DB2_GENERATED           7 ROWID        17      0 N    A   N           -1
     LB3                     8 CLOB          4      0 Y    Y   N           -1
     LB4                     9 CLOB          4      0 Y    Y   N           -1
    -----------------------------------------------------------------------------

   =========

   Object #2

   =========



   Details for package : TRDEL                    in collection : TDKDB


    Package information

     Package type . . . . . . . . . . : Trigger package

     Version  . . . . . . . . . . . . :

     Authorization ID of owner  . . . : TS5807

     Owner type . . . . . . . . . . . : Auth ID

     Authorization ID of creator  . . : TS5807

     Created timestamp  . . . . . . . : 2017-09-28-04.42.49.713457

     Latest BIND timestamp  . . . . . : 2017-09-28-04.42.49.713457

     Version under which package bound: V11

     Qualifier for unqualified SQL  . : TS5807

     Operative status of package  . . : Package is valid and operative

     Resource and authorization check : At BIND time

     Size of the base section (bytes) : 3176   (in EDM pool during execution)

     Average DML section size (bytes) : 19144  (loaded when needed during exec)

     Package bound with EXPLAIN . . . : No

     SQLERROR specified at BIND time  : No - SQLERROR(NOPACKAGE)
 specified
```

Figure 280. Example of a report (Part 2)

```
    BIND or REBIND from remote loc.  : No - (RE)BIND was from a local system
    Remote packages creation method  :
    Source of the package  . . . . . :
    Number of enabled/disabled conn. : 0
    Data concurrency . . . . . . . . : B - not required
      Effect on blocking . . . . . . : Allow blocking for ambiguous cursors
    DEGREE of I/O parallelism  . . . : 1 - parallel I/O inhibited
    Group member that performed BIND :
    Dynamic SQL rules  . . . . . . . : Use binders authid and authorizations
    Re-optimize SQL at execution time: No - access path determined at BIND time
    Defer prepare  . . . . . . . . . : No - do not defer prepare to OPEN time
    Keep prepared dynamic SQL stmts  : No - are destroyed at COMMIT
    Protocol for 3 part names  . . . : D - uses DRDA
    Function resolved at . . . . . . : 2017-09-28-04.42.49.713455
    Optimizer hint identifier  . . . :
    Encode CCSID . . . . . . . . . . : Using EBCDIC default CCSID from install
    Write group buffer pool pages  . : Normal write
    ROUNDING option used on last bind: Round Half Even
    Concurrent Access  . . . . . . . : Not specified - inherit from DB2 ZPARM
    Last date objects used . . . . . : 01/01/0001
    SQL path for resolving UDT,UDF,SP: "SYSIBM","SYSFUN","SYSPROC","SYSIBMAD(*1)

    Precompiler related information:
      Timestamp of precompilation  . : 0001-01-01-00.00.00.000000
      Consistency token in hex . . . : 1A66985A1A9564C3
      SQL escape character . . . . . : ' (apostrophe)
      Decimal point character  . . . : . (period)
      Host program language  . . . . : Remotely bound, trigger, or SQL package
      Mixed character set  . . . . . : N
      Decimal 31 used  . . . . . . . : No
1     Katakana . . . . . . . . . . . : No
```

Figure 281. Example of a report (Part 3)

```
    Resource allocation information:
      Resources are released . . . . : At COMMIT
      Isolation level  . . . . . . . : Cursor stability

    Temporal special register information:
      Sensitive to SYSTEM_TIME . . . : Yes
      Sensitive to BUSINESS_TIME . . : Yes

    Sensitive to GET_ARCHIVE . . . . : Yes

    Bind options:
      Access path reuse behavior . . : No - does not reuse paths
      Package compat level behavior  : V11R1
      Static SQL DESCRIBE requests . : Yes - creates DESCRIBE SQLDA

   SQL statements in package: TDKDB.TRDEL

    SQL in statement: 2      (Stmt id:  4800449 )
      UPDATE TDKDB.TRIGGER SET RECCOUNT = (SELECT COUNT (*) FROM
      TDKDB.TRIGGER, (SELECT * FROM TDKDB.TRIGGER WHERE NAME LIKE '%A%' OR
      NAME LIKE '%Z%') AS B WHERE A.ID = B.ID AND A.NAME = B.NAME) WHERE ID
      LIKE '%124%'
  -------------------------------------------------------------------------------

==========
Object #3
==========

Details for partitioned accelerated table : TS5771.TQMAOTT1

 Partition info  . . : BY_GROWTH
```

Figure 282. Example of a report (Part 4)

```
Part no : 1
 Logical no  . . . . : 1
 Limit key value . . :


 Change information  :
   Category  . . . . : NONE
   Last load TS  . . : 1970-01-01T00:00:00.000000Z
   Type  . . . . . . : AcceleratorOnly
   Shared tablespace : false
-------------------------------------------------------------------------------
==========
Object #4
==========
Details for non-partitioned accelerated table : TS5771A.TBAD0099

 Change information  :
   Category  . . . . : UNKNOWN
   Last load TS  . . : 2017-06-26T22:02:21.713980Z
   Type  . . . . . . : DataChange
   Shared tablespace : true
------------------------------------------------------------------------------
  Long names legend
  (*1) - "SYSIBM","SYSFUN","SYSPROC","SYSIBMADM","TS5807"

=================================
End of Details for Objects Report
=================================
```

Figure 283. Example of a report (Part 5)

Retrieving details about tables in batch mode

You can use batch mode to retrieve details about tables.

About this task

Procedure

1. In the **Option** field on the **Administration Menu** panel, specify 1, and press Enter.
   The **System Catalog** panel is displayed.
2. Specify T, and press Enter.
   The **Tables, Views, and Aliases** panel is displayed.
3. Find the table whose details you want to retrieve in the **Name** field.
4. Choose one of the following methods:
   - To retrieve details about a single table, specify BET in the **Sel** column next to the object, and press Enter.

- To retrieve details about all the tables, issue the BET command in the Command line, and press Enter.

The JCL is generated. For more information, see Figure 276 on page 508.

The PARMS section shows parameter usage for tables, as shown in the following example:

```
//PARMS    DD *
TYPE='TB' QUAL='SYSIBM' NAME='DSN_PREDICAT_TABLE';
TYPE='TB' QUAL='SYSIBM' NAME='DSN_PREDICATE_SELECTIVITY';
TYPE='TB' QUAL='DSN8BQRY' NAME='DSN_PREDICAT_TABLE';
TYPE='TB' QUAL='DSN8BQRY' NAME='DSN_PREDICATE_SELECTIVITY';
```

5. Submit the job.

Retrieving details about packages in batch mode

You can use batch mode to retrieve details about packages.

Procedure

1. In the **Option** field on the **Administration Menu** panel, specify 1, and press Enter.

   The **System Catalog** panel is displayed.
2. Specify K, and press Enter.

   The **Packages** panel is displayed.
3. Find the package whose details you want to retrieve in the **Name** field.
4. Choose one of the following methods:

   - To retrieve details about a single package, specify BET in the **Sel** column next to the object, and press Enter.
   - To retrieve details about all the packages, issue the BET command in the Command line, and press Enter.

   The JCL is generated. For more information, see Figure 276 on page 508.

   The PARMS section shows parameter usage for packages, as shown in the following example:

```
//PARMS    DD *
TYPE='PK' QUAL='CKZPBPK' NAME='CKZ00716' VERSION='';
TYPE='PK' QUAL='CKZPBPK' NAME='CKZ00718' VERSION='';
```

5. Submit the job.

Retrieving details about accelerated tables in batch mode

You can use batch mode to retrieve details about tables.

Procedure

1. In the **Option** field on the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify AT, and press Enter.

   The **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel is displayed.
3. Specify 1, and press Enter.

   The **Display Accelerated Tables (ADBPZAT)** panel is displayed.
4. Find the accelerated table whose details you want to retrieve in the **Name** field.
5. Choose one of the following methods:

   - To retrieve details about a single accelerated table, specify BET in the **Sel** column next to the object, and press Enter.

- To retrieve details about all the accelerated tables, issue the BET command in the Command line, and press Enter.

The JCL is generated. For more information, see Figure 276 on page 508.

The PARMS section shows parameter usage for accelerated tables, as shown in the following example:

```
//PARMS    DD *
TYPE='AT' QUAL='HLDEFLT' NAME='DEFAULTIF_N_LOAD' ACCELERATOR='IDAAZ01';
TYPE='AT' QUAL='HLDEFLT' NAME='DEFAULT_FIELDS' ACCELERATOR='IDAAZ01';
TYPE='AT' QUAL='HLDB0001' NAME='HLDB01T1' ACCELERATOR='IDAAZ01';
TYPE='AT' QUAL='HLDB0008' NAME='HLDB01T3' ACCELERATOR='IDAAZ01';
TYPE='AT' QUAL='HLDB0009' NAME='HLDB01T3' ACCELERATOR='IDAAZ01';
TYPE='AT' QUAL='HLDISC' NAME='HLDISC8T' ACCELERATOR='IDAAZ01';
TYPE='AT' QUAL='HLDEFLT' NAME='NULLABLE_FIELDS' ACCELERATOR='IDAAZ01';
TYPE='AT' QUAL='HLDEFLT' NAME='NULLIF_IN_LOAD' ACCELERATOR='IDAAZ01';
TYPE='AT' QUAL='HLDBARCA' NAME='TBARC1' ACCELERATOR='IDAAZ01';
TYPE='AT' QUAL='HLDBARCB' NAME='TBARC1' ACCELERATOR='IDAAZ01';
```

6. Submit the job.

Requesting revoke impact reports in batch

A *revoke impact report* helps you determine how the authorizations and database objects will be affected by revoking an authorization before you actually revoke it. You can request these reports on the **Revoke *object* Privileges** panels. You can also request these reports by using a batch job.

About this task

A sample job for requesting revoke impact reports is provided in member ADBRIPB in the sample library. This job runs a REXX exec that invokes the revoke impact program for each request that is specified in the job. This job consists of the following parts:

- A JOB statement
- An in-line procedure with parameters (`//ADBRIPB PROC`)
- A step that invokes the in-line procedure (`// EXEC ADBRIPB`)

Procedure

To request revoke impact reports in batch:

1. Copy the job in SAMP(ADBRIPB) and edit it as needed:

   - Edit the JOB statement as needed.
   - In ADBRIPB PROC statement, modify the following parameter values for your system:

     PROD
     The high-level qualifier (HLQ) for the Db2 Admin Tool libraries.

     DSNEXIT
     The data set name (DSN) of the Db2 DSNEXIT library.

     DSNLOAD
     The DSN of the Db2 DSNLOAD library.

     ISPHLQ
     The HLQ of the ISPF libraries.

     SSID
     The SSID of the Db2 subsystem to use.

     PLAN
     The plan name for the revoke impact program.

     DEBUG
     A parameter to enable traces for serviceability.

- If you use names other than the standard SMP/E library names for the Db2 Admin Tool and ISPF libraries, update the in-line procedure with the names you are using. The provided in-line procedure uses the standard names for these libraries.
- In PARMS DD set **DB2REL** to your Db2 release. Use the format *VVRM*.
- In REQS DD, modify the following parameters as needed to specify your request:

  TYPE
  The type of object. Use the values on the **System Catalog (ADB21)** panel. For example, specify S for table space.

  QUAL
  An optional qualifier for the object.

  NAME
  The name of the object.

  GRANTEES
  A list of grantees to be revoked. If more than one grantee is specified, separate the grantees by commas and enclose the entire list in single quotation marks (').

  PRIV
  A list of privileges to revoke. If more than one privilege is specified, separate the privileges by commas and enclose the entire list in single quotation marks (').

  BY
  Authid for the BY clause of the REVOKE statement.

  IDP
  Whether to include dependent privileges (the INCLUDING DEPENDENT PRIVILEGES clause in the REVOKE statement). Possible values are Y or N.

  Each request can span multiple lines and must be terminated with a semicolon. For example:

  ```
  TYPE=T, QUAL=SYSIBM, NAME=SYSDUMMY1, GRANTEES=’USER1,USER2’,PRIV=’SELECT’,BY=SYSADM,IDP=Y;
  ```

  You can specify multiple requests in REQS DD. Separate each request by a semicolon.

  You can also specify common parameters for all requests in PARMS DD. (Any of the parameters in the preceding list for REQS DD can be specified in PARMS DD as a common parameter.) For example, if TYPE is always T and the qualifier is always D1234, you can specify the following:

  ```
  //PARMS DD *
  DB2AREL=1310,TYPE=T, QUAL=D1234;
  //REQS DD *
  NAME=T1, ...;
  NAME=T2, ...;
  ```

  This specification is functionally equivalent to the following:

  ```
  //PARMS DD *
  DB2AREL=1310;
  //REQS DD *
  TYPE=T, QUAL=D1234,NAME=T1, ...;
  TYPE=T, QUAL=D1234,NAME=T2, ...;
  ```

2. Submit the job.

   Db2 Admin Tool generates a report similar to the following example:

   ```
   Parameters on DDNAME:PARMS
   DB2REL=1215
   IDP=Y
   Parameters on DDNAME:REQS
   TYPE=T
   QUAL=TUJTEST4
   NAME=MYTAB
   GRANTEES=YYY,XXXXXX
   PRIV=SELECT,DELETE,INSERT
   BY=TS5816
   ```

```
Report:
                          Owner/
Grantee  G Resource N  O  Schema/  Grantor/ G H Privileges/
         T Collection  T  P/K Name Binder   T G Effect
-------- - ----------- -- -------- -------- - - -----------
XXXXXX     MYTAB       T  TUJTEST4 TS5816           G
End of report
```

Migrating Db2 objects, data, views, and catalog statistics

Db2 Admin Tool enables you to *migrate*, or copy, Db2 object definitions, the data for the objects, views, and the catalog statistics for the objects from one Db2 subsystem to other Db2 subsystems.

About this task

You can migrate any combination of this set of information (object definitions, data, views, and statistics) for Db2 databases, table spaces, and tables, as well as their dependent objects. When you migrate information, Db2 Admin Tool attempts to preserve as many of the dependent definitions as possible, such as indexes, views, table checks, synonyms, aliases, and authorizations on these objects.

Typically, this migrate (MIG) function is used to perform the following tasks:

- Create a separate Db2 test system
- Move a test system into a production system
- Copy statistics from a production system to a development (or test) system so that you can test new and modified programs with the statistics from the production system
- Consolidate two separate database systems into one

You can request either that Db2 Admin Tool migrate the objects or that Db2 Cloning Tool clone the objects.

Restrictions: The following restrictions apply to this migration process:

- When a schema is associated with a database, you must migrate the database structure and the schema separately.
- Databases without table spaces are not migrated. Db2 Admin Tool issues a warning message that no rows are returned.
- For table spaces that are created with the DEFINE NO option or that are empty (or tables within these table spaces), you can migrate only the schema definitions (DDL). JCL or statements to unload the data are not created during migration.
- When migrating at the table level and the table has a LOB column, if the migrate option **DROP on target before create** is set to Yes, the base table and any auxiliary LOB tables are dropped. Any LOB table space that was explicitly defined is not dropped. You must manually drop the LOB table spaces.
- If the base table containing one or more LOB columns is dropped and recreated, the explicit auxiliary table is recreated according to its source definition. Changes to the auxiliary table are not reported. Updates to the auxiliary table are ignored if the base table is not recreated.
- If you migrate the catalog statistics for the objects, the statistics for materialized query tables are not included.

Procedure

To migrate Db2 objects, data, views, and catalog statistics:

1. “Step 1. Specify the objects and information to migrate” on page 520
2. “Step 2. Generate the migration batch jobs” on page 522
3. “Step 3. Run the migration batch jobs” on page 525
4. “Step 4. Optional: Transfer the jobs, work statement list, and data to the target system” on page 526
5. “Step 5. Run the batch define, reload, and optional jobs” on page 527

Related information

“The Batch Restart programs: ADBTEP2 and ADBTEPA” on page 572
The Batch Restart program, ADBTEP2, and the Authorization Switching program, ADBTEPA, are used with work statement lists and the Alter and Migrate Db2 data functions.

Step 1. Specify the objects and information to migrate

You can migrate object definitions, object data, views, catalog statistics, or any combination of this information for Db2 databases, table spaces, and tables, as well as their dependent objects.

Procedure

To specify the objects and information to migrate:

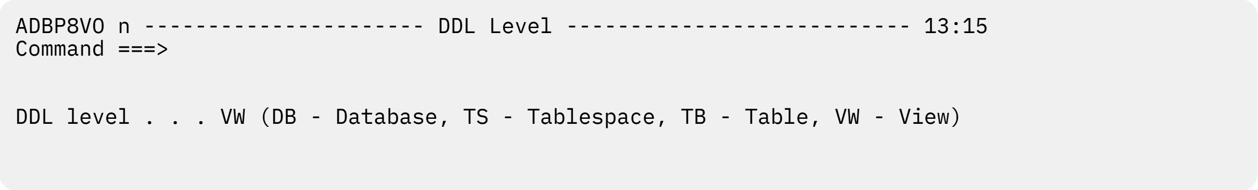
1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify one of the following options, and press Enter:

   - D for databases
   - S for table spaces
   - T for tables
   - V for views

   You can use any of these object types as a base to select objects that you want to migrate.
3. On the **Databases (ADB21D)** panel, **Table Spaces (ADB21S)** panel, **Tables, Views, and Aliases (ADB21T)** panel, or **Views (ADB21VV)** panel, issue the MIG primary command or line command.

   Tip: If you use the MIG primary command, first filter the list of objects on the panel so that only the objects that you want to migrate are listed.
4. On the **Migrate Table Spaces (ADB28S)** panel, **Migrate Tables (ADB28T)** panel, or **Migrate Views (ADBP8V)** panel, take one of the following actions:

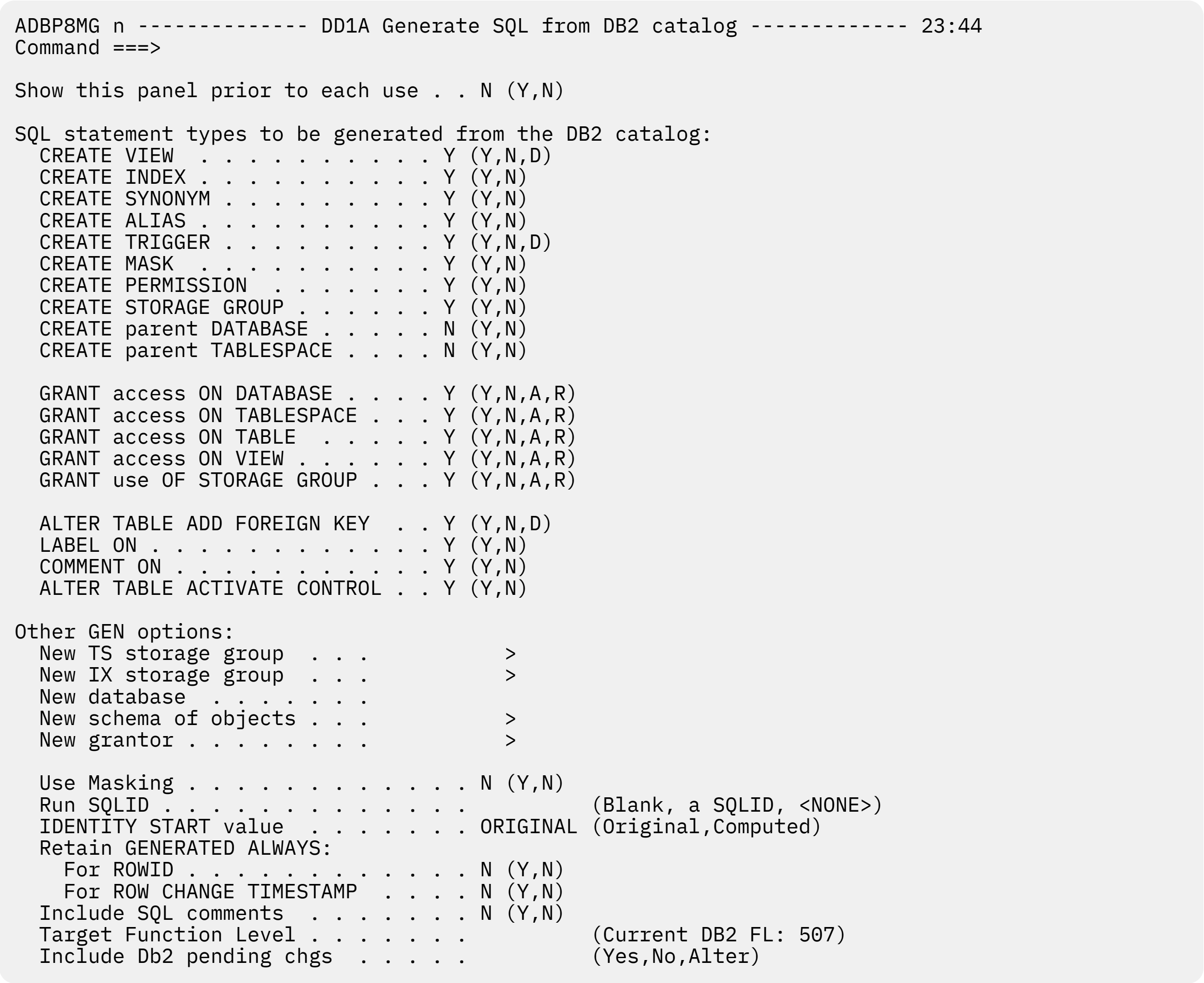
   - To specify the objects individually, modify the list of objects on this panel as needed by using the available commands and line commands. For example, on the **Migrate Table Spaces (ADB28S)** panel, you can use the MIGARC command to add all related archive table spaces.
   - To specify an object scope, do not modify the list of objects on this panel. You can later specify the object scope as part of “Step 2. Generate the migration batch jobs” on page 522.
5. Specify the NEXT command.
6. For views, specify the level at which you want to migrate the view on the **DDL Level (ADBP8VO)** panel, and press Enter.



```
ADBP8VO n --------------------- DDL Level -------------------------- 13:15
Command ===>


DDL level . . . VW (DB - Database, TS - Tablespace, TB - Table, VW - View)
```

*Figure 284. **DDL Level (ADBP8VO)** panel*

7. On the **Migrate Parameters (ADB28M)** panel, specify the GEN option, and press Enter.
8. On the **Generate SQL from DB2 catalog (ADBP8MG)** panel, specify the SQL statements that you want generated for the migration, and press Enter:



```
ADBP8MG n ------------- DD1A Generate SQL from DB2 catalog ------------ 23:44
Command ===>

Show this panel prior to each use . . N (Y,N)

SQL statement types to be generated from the DB2 catalog:
  CREATE VIEW  . . . . . . . . . . Y (Y,N,D)
  CREATE INDEX . . . . . . . . . . Y (Y,N)
  CREATE SYNONYM . . . . . . . . . Y (Y,N)
  CREATE ALIAS . . . . . . . . . . Y (Y,N)
  CREATE TRIGGER . . . . . . . . . Y (Y,N,D)
  CREATE MASK  . . . . . . . . . . Y (Y,N)
  CREATE PERMISSION  . . . . . . . Y (Y,N)
  CREATE STORAGE GROUP . . . . . . Y (Y,N)
  CREATE parent DATABASE . . . . . N (Y,N)
  CREATE parent TABLESPACE . . . . N (Y,N)

  GRANT access ON DATABASE . . . . Y (Y,N,A,R)
  GRANT access ON TABLESPACE . . . Y (Y,N,A,R)
  GRANT access ON TABLE  . . . . . Y (Y,N,A,R)
  GRANT access ON VIEW . . . . . . Y (Y,N,A,R)
  GRANT use OF STORAGE GROUP . . . Y (Y,N,A,R)

  ALTER TABLE ADD FOREIGN KEY  . . Y (Y,N,D)
  LABEL ON . . . . . . . . . . . . Y (Y,N)
  COMMENT ON . . . . . . . . . . . Y (Y,N)
  ALTER TABLE ACTIVATE CONTROL . . Y (Y,N)

Other GEN options:
  New TS storage group  . . .           >
  New IX storage group  . . .           >
  New database  . . . . . . .
  New schema of objects . . .           >
  New grantor . . . . . . . .           >

  Use Masking . . . . . . . . . . . . N (Y,N)
  Run SQLID . . . . . . . . . . . . .          (Blank, a SQLID, <NONE>)
  IDENTITY START value  . . . . . . . ORIGINAL (Original,Computed)
  Retain GENERATED ALWAYS:
    For ROWID . . . . . . . . . . . . N (Y,N)
    For ROW CHANGE TIMESTAMP  . . . . N (Y,N)
  Include SQL comments  . . . . . . . N (Y,N)
  Target Function Level . . . . . . .          (Current DB2 FL: 507)
  Include Db2 pending chgs  . . . . .          (Yes,No,Alter)
```

Figure 285. ***Generate SQL from DB2 catalog (ADBP8MG)*** *panel*

9. On the **Migrate Parameters (ADB28M)** panel, specify the following options to indicate which information you want to migrate:

   Scope of migrate:

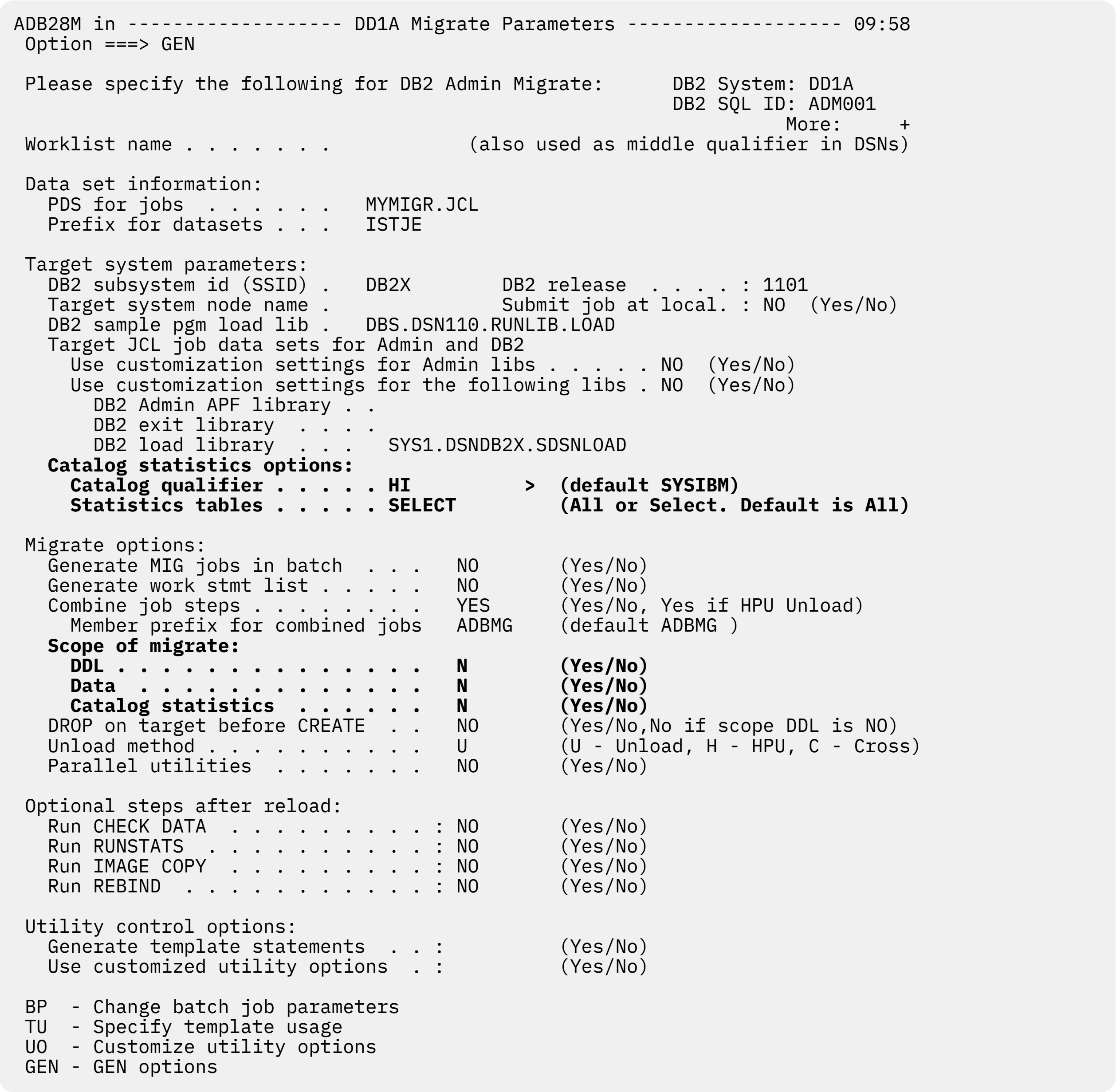
   Specify whether you want to migrate DDL, data, and catalog statistics.

   Catalog statistics options:

   Specify the qualifier for the target catalog and the catalog tables that you want to update.

   When you migrate catalog statistics, Db2 Admin Tool generates INSERT, UPDATE, and DELETE statements that modify the catalog statistics. The statements are generated with the qualifier of the target catalog that you specify. The statistic fields that are modified are those fields that are associated with the objects that are being migrated. (The complete list of statistics fields are those fields that are set by RUNSTATS that can be modified and the five statistics columns for table functions in SYSROUTINES, which are not set by RUNSTATS.)

   Tip: To control the number of generated statistics, specify SELECT in the **Statistics tables** field. Later in the process (as part of "Step 2. Generate the migration batch jobs" on page 522), you can select which catalog tables you want to update with statistics.



```
ADB28M in ------------------- DD1A Migrate Parameters ------------------- 09:58
 Option ===> GEN

 Please specify the following for DB2 Admin Migrate:     DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
                                                                   More:     +
 Worklist name . . . . . . .            (also used as middle qualifier in DSNs)

 Data set information:
   PDS for jobs  . . . . . .   MYMIGR.JCL
   Prefix for datasets . . .   ISTJE

 Target system parameters:
   DB2 subsystem id (SSID) .   DB2X        DB2 release  . . . . : 1101
   Target system node name .               Submit job at local. : NO  (Yes/No)
   DB2 sample pgm load lib .   DBS.DSN110.RUNLIB.LOAD
   Target JCL job data sets for Admin and DB2
     Use customization settings for Admin libs . . . . . NO  (Yes/No)
     Use customization settings for the following libs . NO  (Yes/No)
       DB2 Admin APF library . .
       DB2 exit library  . . . .
       DB2 load library  . . .   SYS1.DSNDB2X.SDSNLOAD
   Catalog statistics options:
     Catalog qualifier . . . . . HI          >  (default SYSIBM)
     Statistics tables . . . . . SELECT         (All or Select. Default is All)

 Migrate options:
   Generate MIG jobs in batch  . . .   NO       (Yes/No)
   Generate work stmt list . . . . .   NO       (Yes/No)
   Combine job steps . . . . . . . .   YES      (Yes/No, Yes if HPU Unload)
     Member prefix for combined jobs   ADBMG    (default ADBMG )
   Scope of migrate:
     DDL . . . . . . . . . . . . . .   N        (Yes/No)
     Data  . . . . . . . . . . . . .   N        (Yes/No)
     Catalog statistics  . . . . . .   N        (Yes/No)
   DROP on target before CREATE  . .   NO       (Yes/No,No if scope DDL is NO)
   Unload method . . . . . . . . . .   U        (U - Unload, H - HPU, C - Cross)
   Parallel utilities  . . . . . . .   NO       (Yes/No)

 Optional steps after reload:
   Run CHECK DATA  . . . . . . . . . : NO       (Yes/No)
   Run RUNSTATS  . . . . . . . . . . : NO       (Yes/No)
   Run IMAGE COPY  . . . . . . . . . : NO       (Yes/No)
   Run REBIND  . . . . . . . . . . . : NO       (Yes/No)

 Utility control options:
   Generate template statements  . . :          (Yes/No)
   Use customized utility options  . :          (Yes/No)

 BP  - Change batch job parameters
 TU  - Specify template usage
 UO  - Customize utility options
 GEN - GEN options
```

Figure 286. ***Migrate Parameters (ADB28M)*** *panel*

Keep this panel open. In the next step ("Step 2. Generate the migration batch jobs" on page 522), you will continue entering options on the **Migrate Parameters (ADB28M)** panel.

What to do next

"Step 2. Generate the migration batch jobs" on page 522

Step 2. Generate the migration batch jobs

You can choose to have the migration batch jobs generated online or in batch. Generate the jobs in batch if you want to specify an object scope.

If you specify to have the migration jobs generated in batch, Db2 Admin Tool creates a work data set (MIGVARS) that stores the parameter information that you specify on **Migrate Parameters (ADB28M)** panel and the necessary ISPF tables to use as input when generating the migration source and target JCL. For more information about the data sets that are required during this migration process, see "Work data sets used by the MIG function" on page 527.

Before you begin

Before you generate the migration batch jobs, complete "Step 1. Specify the objects and information to migrate" on page 520.

Procedure

To generate the migration batch jobs:

1. On the **Migrate Parameters (ADB28M)** panel, specify the following job and data set options, and press Enter:

   Worklist name and Data set information:
   Use these fields to change the default qualifier values that are used for the MIGVARS data set.

   Target system parameters:
   Specify the SSID, node name, sample program load library, and JCL data set information for the target subsystem.

   Migrate options:
   Specify the remaining options (those options other than **Scope of migrate**, which you specified in the last step in "Step 1. Specify the objects and information to migrate" on page 520).

   If you want to specify an object scope, specify YES for the **Generate MIG jobs in batch** option.

   If you specify YES for the **DROP on target before CREATE** field, any RESTRICT ON DROP conditions for tables are not considered. If a table is defined with RESTRICT ON DROP, you must remove that attribute for the DROP statement to complete successfully.

   If you specify UNLOAD in the **Unload method** field and YES in the **Parallel utilities** field, and do not provide your own UNLDDN template, the default template ASYREC6 with variable &PART or &PA in the ADB2UCUS skeleton is used as the template for the unload data set. When &PART or &PA is specified, Db2 Admin Tool replaces the variable with 00001 up to the maximum partition number of the associated object. The total length of the values for &PREFIX and &LEVEL must not exceed 12 bytes.

   Optional steps after reload:
   Specify which jobs you want to run after reloading the data.

   The CHECK DATA, RUNSTATS, and COPY requests are not generated for implicitly created table spaces.

   The REBIND option generates rebinds of the source packages for the target system.

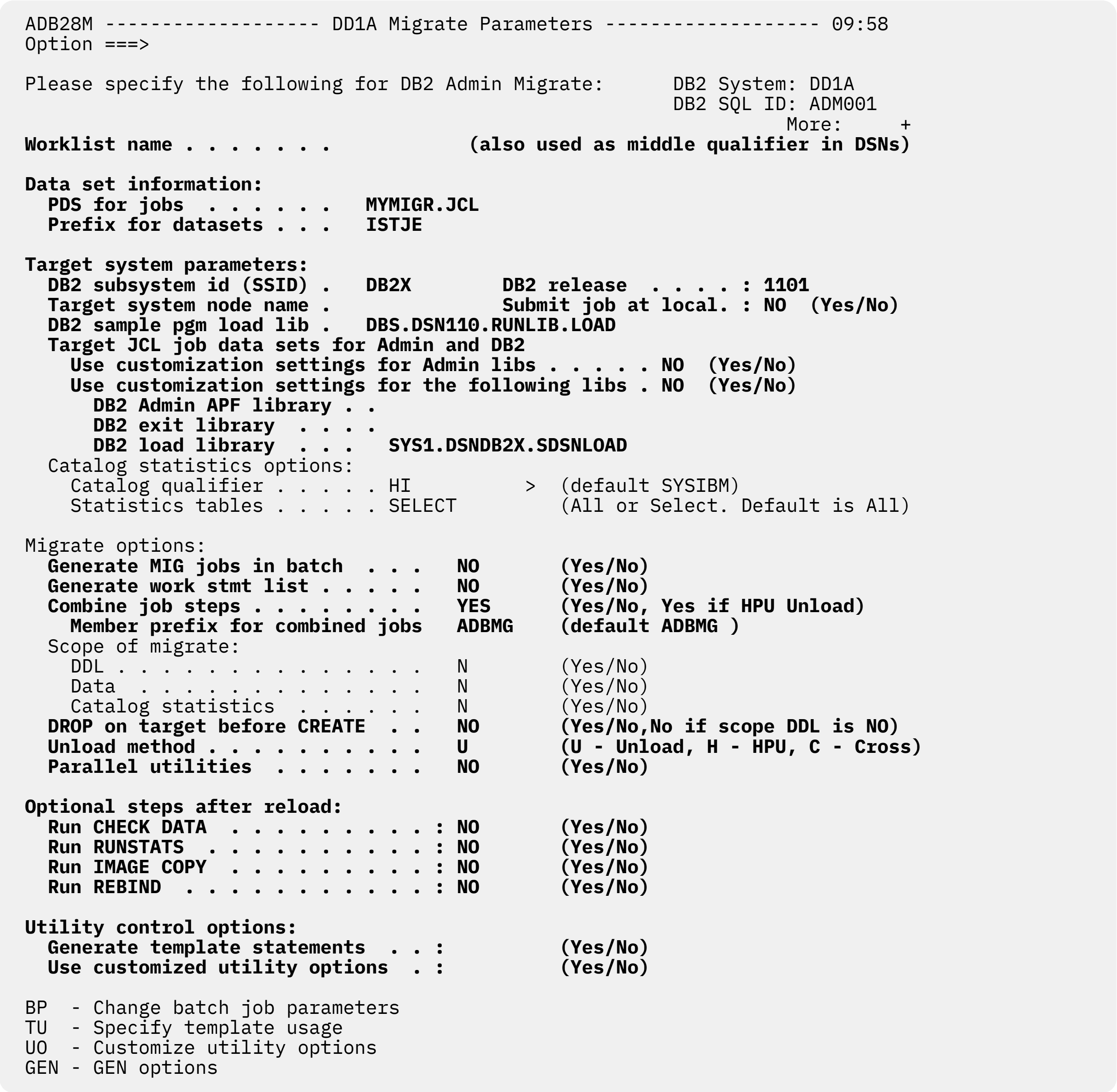
   Utility control options
   Specify whether to use templates and custom utility options.

   If you set the option **Use customized utility options** to YES, use the UO (Customize utility options) command to specify the utility options. To customize the HPU utility, enter the HPU option on the **Change Utilities Options (ADB2UOPS)** panel, as described in "Unloading objects by using Db2 High Performance Unload " on page 613.

   If you choose to migrate only the data, use the LOAD utility option REPLACE or RESUME to control how the data is loaded into the target system with the following restrictions:

   - Db2 does not allow using LOAD REPLACE on certain types of tables. When LOAD REPLACE is specified but not allowed, the RESUME YES option is used instead.
   - If all tables in a multi-table table space are not selected for migration on the source system, and the LOAD REPLACE option is specified, RESUME YES is used instead.
   - If all tables in a multi-table table space are selected for migration on the source system, and the LOAD REPLACE option is specified, any additional tables in the table space on the target system remain empty after migration.

   If you do not use customized utility options, Db2 Admin Tool uses the default REPLACE and RESUME option that is generated by the Db2 UNLOAD utility or the Db2 High Performance Unload (HPU) utility.



```
ADB28M ------------------- DD1A Migrate Parameters ------------------- 09:58
Option ===>

Please specify the following for DB2 Admin Migrate:      DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
                                                                   More:     +
Worklist name . . . . . . .           (also used as middle qualifier in DSNs)

Data set information:
  PDS for jobs  . . . . . .   MYMIGR.JCL
  Prefix for datasets . . .   ISTJE

Target system parameters:
  DB2 subsystem id (SSID) .   DB2X        DB2 release  . . . . : 1101
  Target system node name .               Submit job at local. : NO  (Yes/No)
  DB2 sample pgm load lib .   DBS.DSN110.RUNLIB.LOAD
  Target JCL job data sets for Admin and DB2
    Use customization settings for Admin libs . . . . . NO  (Yes/No)
    Use customization settings for the following libs . NO  (Yes/No)
      DB2 Admin APF library . .
      DB2 exit library  . . . .
      DB2 load library  . . .   SYS1.DSNDB2X.SDSNLOAD
  Catalog statistics options:
    Catalog qualifier . . . . . HI          >  (default SYSIBM)
    Statistics tables . . . . . SELECT         (All or Select. Default is All)

Migrate options:
  Generate MIG jobs in batch  . . .   NO       (Yes/No)
  Generate work stmt list . . . . .   NO       (Yes/No)
  Combine job steps . . . . . . . .   YES      (Yes/No, Yes if HPU Unload)
    Member prefix for combined jobs   ADBMG    (default ADBMG )
  Scope of migrate:
    DDL . . . . . . . . . . . . . .   N        (Yes/No)
    Data  . . . . . . . . . . . . .   N        (Yes/No)
    Catalog statistics  . . . . . .   N        (Yes/No)
  DROP on target before CREATE  . .   NO       (Yes/No,No if scope DDL is NO)
  Unload method . . . . . . . . . .   U        (U - Unload, H - HPU, C - Cross)
  Parallel utilities  . . . . . . .   NO       (Yes/No)

Optional steps after reload:
  Run CHECK DATA  . . . . . . . . . : NO       (Yes/No)
  Run RUNSTATS  . . . . . . . . . . : NO       (Yes/No)
  Run IMAGE COPY  . . . . . . . . . : NO       (Yes/No)
  Run REBIND  . . . . . . . . . . . : NO       (Yes/No)

Utility control options:
  Generate template statements  . . :         (Yes/No)
  Use customized utility options  . :         (Yes/No)

BP  - Change batch job parameters
TU  - Specify template usage
UO  - Customize utility options
GEN - GEN options
```

Figure 287. ***Migrate Parameters (ADB28M)*** *panel*

After you press Enter, Db2 Admin Tool starts to generate the jobs required for migration.

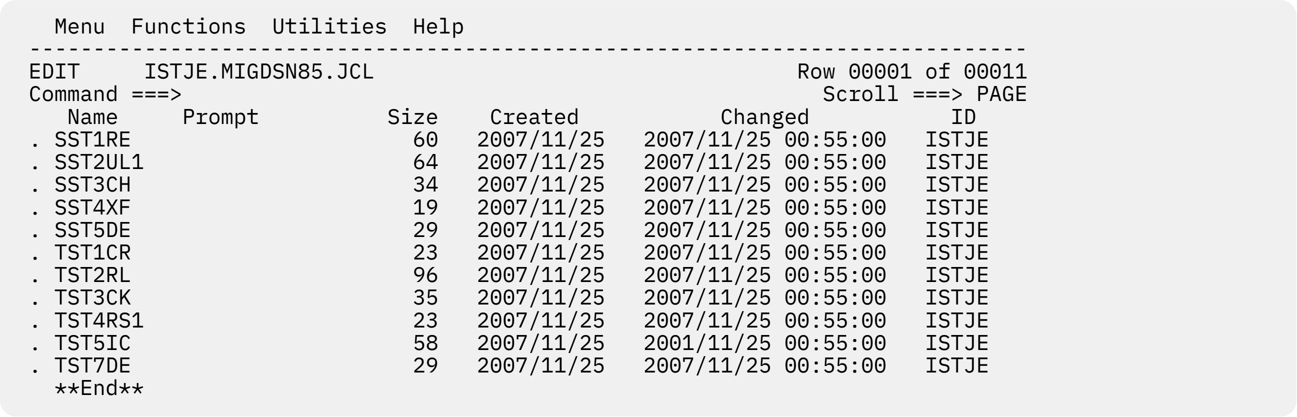
2. If the **Specify Masks (ADB2GENM)** panel is displayed, follow the instructions in "Specifying masks" on page 315.
3. If the **Catalog Statistics Tables (ADBPGEN2)** panel is displayed (because you specified SELECT in the **Statistics tables** field), specify the catalog tables that you want to update with statistics. Remove the slash (**/**) next to catalog tables that you do not want to update.

   For example, removing the slash (**/**) that is next to SYSCOLDISTSTATS turns off the generation of updates to that table.

   Tip: Tables with the suffix of DISTSTATS are used to store partition-level statistics. These tables are not used by the optimizer but are used by RUNSTATS. Therefore, tables with the suffix of DISTSTATS can be turned off if you do not plan to run RUNSTATS on the target objects.

If you chose to generate the migration jobs online, the **Migrate Progress (ADB28MP)** panel is displayed to show the progress of building the jobs. In this case, you do not need to take any action. When Db2 Admin Tool finishes generating the jobs that are required for migration, they are displayed in an ISPF

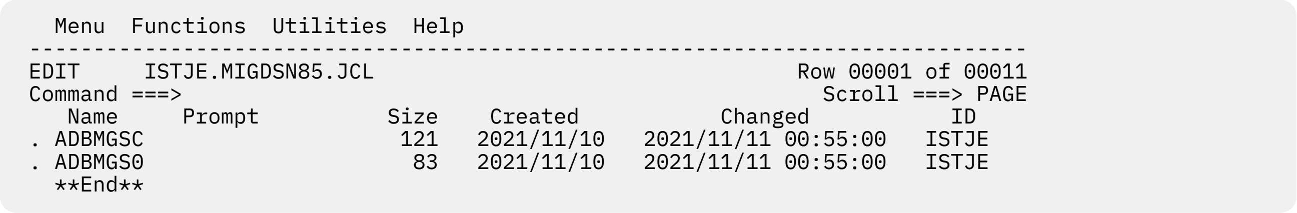
Edit session, as shown in the following example. You can skip the remaining steps in this procedure and continue to Step 3. Run the migration batch jobs.



```
  Menu  Functions  Utilities  Help
------------------------------------------------------------------------------
EDIT     ISTJE.MIGDSN85.JCL                                 Row 00001 of 00011
Command ===>                                                  Scroll ===> PAGE
   Name     Prompt          Size    Created          Changed          ID
. SST1RE                      60   2007/11/25   2007/11/25 00:55:00   ISTJE
. SST2UL1                     64   2007/11/25   2007/11/25 00:55:00   ISTJE
. SST3CH                      34   2007/11/25   2007/11/25 00:55:00   ISTJE
. SST4XF                      19   2007/11/25   2007/11/25 00:55:00   ISTJE
. SST5DE                      29   2007/11/25   2007/11/25 00:55:00   ISTJE
. TST1CR                      23   2007/11/25   2007/11/25 00:55:00   ISTJE
. TST2RL                      96   2007/11/25   2007/11/25 00:55:00   ISTJE
. TST3CK                      35   2007/11/25   2007/11/25 00:55:00   ISTJE
. TST4RS1                     23   2007/11/25   2007/11/25 00:55:00   ISTJE
. TST5IC                      58   2007/11/25   2001/11/25 00:55:00   ISTJE
. TST7DE                      29   2007/11/25   2007/11/25 00:55:00   ISTJE
  **End**
```

Figure 288. Sample migrate edit panel

If you chose to generate the jobs in batch, an ISPF edit session opens and lists the generated jobs. These jobs are needed to generate the migration jobs. In this case, continue with the remaining steps.



```
  Menu  Functions  Utilities  Help
------------------------------------------------------------------------------
EDIT     ISTJE.MIGDSN85.JCL                                 Row 00001 of 00011
Command ===>                                                  Scroll ===> PAGE
   Name     Prompt          Size    Created          Changed          ID
. ADBMGSC                    121   2021/11/10   2021/11/11 00:55:00   ISTJE
. ADBMGS0                     83   2021/11/10   2021/11/11 00:55:00   ISTJE
  **End**
```

Figure 289. Sample of job edit panel for generating the migrate jobs in batch

4. If you want to specify an object scope, edit one of the following jobs according to the instructions in the job comments:

   - *<Member prefix for combined jobs>*SC, if you chose to combine the job steps
   - SSTSCBAT, if you chose not to combine the job steps

   This job redefines the object scope that is saved in the ADBSVARS data set for generating the migration batch jobs.

5. Submit one of the following jobs to generate the jobs that are required for migration:

   - *<Member prefix for combined jobs>*S0, if you chose to combine the job steps
   - SST0BAT, if you chose not to combine the job steps

What to do next

"Step 3. Run the migration batch jobs" on page 525

Step 3. Run the migration batch jobs

As part of the process of migrating Db2 object definitions, object data, views, and statistics, you need to run the migration jobs that were generated by Db2 Admin Tool.

Before you begin

Before you run the migration batch jobs, complete "Step 2. Generate the migration batch jobs" on page 522

Procedure

To run the migration batch jobs:

1. Review the following source system jobs and submit them in the listed sequence:

   a. SST1RE - Performs reverse engineering.

   b. SST2UL*n* - Unloads data; *n* is an integer. If you are migrating many table spaces, multiple unload jobs might be created.

   c. SST3CH - Changes unload control data sets.

2. Run the batch jobs as follows:

   - If you combined the job steps, these jobs are located in the group xxxxS1. Run the first group with the name *xxxx*S1 on the source system.
   - If you specified the current system node name to be the name of the target system node, the source and target systems are the same. In this case, run all of the generated jobs on the same system and skip "Step 4. Optional: Transfer the jobs, work statement list, and data to the target system" on page 526.
   - If you requested that a work statement list (WSL) be generated and are running in local mode (that is, not connected to a remote subsystem), run the *xxx*S*n* job in sequence to extract the DDL, unload the data, change the LOAD control statements, and write the work statement list.
   - If you are running in DRDA mode (that is, connected to a remote system), run the *xxx*Sn job first to unload the data on the remote (source) system. After the *xxx*S*n* jobs are complete and the data sets contain the unloaded data and the LOAD control statements are transferred from the remote system to the local system, run the *xxx*L*n* job to extract the DDL, change the LOAD control statements, and write the work statement list.

     Usually, only one *xxx*S*n* job exists to unload the data. However, if many table spaces require unloading, multiple *xxx*S*n* jobs are generated. The final *xxx*S*n* job on the remote system specifies the data set names that need to be transferred to the local system for creating the WSL. n in the *xxx*L*n* job is one greater than the *n* in the last *xxx*S*n* job.

   Tip: If one or more SQL statements fail when you run a migrate job, you can use the Batch Restart program (ADBTEP2) to restart or resume the job at an intermediate point.

What to do next

"Step 4. Optional: Transfer the jobs, work statement list, and data to the target system" on page 526
or
"Step 5. Run the batch define, reload, and optional jobs" on page 527

Step 4. Optional: Transfer the jobs, work statement list, and data to the target system

After you have run the batch jobs, perform this step only if the source and target systems are different; that is, the node names for the source and target systems are not the same.If the source and target database systems are on separate machines, you might need to transfer the information electronically or by using a portable medium, such as a tape.

Before you begin

Before you transfer the jobs, work statement list (WSL), and data to the target system, complete "Step 3. Run the migration batch jobs" on page 525

Procedure

Run the following jobs in the listed sequence:

a. SST4XF - Information about the data sets that needs to be transferred

b. SST5DE - Delete data sets on source system

If you combined job steps, these jobs are located in the group *xxxx*SE. Run the second group with the name *xxxx*SE on the source system after all jobs in the first group are complete.

If your source and target Db2 subsystems are on the same machine, do not run the delete data sets on the source system job (SST5DE or xxxxSE) until you run all of the jobs for the target system.

If you requested that a WSL be generated, the job name *xxx*SE is used (when not connected to a remote system). Otherwise, the name *xxx*LE is used. This job specifies the data set names with the WSL that is required to be transferred to the target system, along with a job step to delete the data sets. Do not run the step to delete the data sets if you are using the WSL.

What to do next

"Step 5. Run the batch define, reload, and optional jobs" on page 527

Step 5. Run the batch define, reload, and optional jobs

The final step in the migration process is to run the batch define, reload, and optional jobs.

Before you begin

Before you run the batch define, reload, and optional jobs, complete "Step 3. Run the migration batch jobs" on page 525.

Procedure

Review the following target system jobs and submit them in the listed sequence:

a. TST1CR - Creates objects and changes the catalog statistics (updates, inserts, and deletes) on the target system.
b. TST2RL*n* - Reloads data; *n* is an integer. If many table spaces are being reloaded, multiple reload jobs can be created.
c. TST3CK - Runs the CHECK DATA utility (optional).
d. TST4RS - Runs the RUNSTATS utility (optional).
e. TST5IC - Takes an image copy (optional).
f. TST6RB - Rebinds (optional).
g. TST7DE - Deletes data sets on the target system.

If you combined job steps, these jobs are located in the group *xxxx*T1. Run group *xxxx*T1 on the target system. If you performed "Step 4. Optional: Transfer the jobs, work statement list, and data to the target system" on page 526, ensure that all the jobs from group *xxxx*SE are complete before running group *xxxx*T1.

If you specified the current system node name to be the name of the target system node, the source and target systems are the same. In this case, run these jobs for Step 5 on the same system as the jobs that you ran for the source system in "Step 3. Run the migration batch jobs" on page 525.

Work data sets used by the MIG function

The Db2 Admin Tool migrate (MIG) function creates and uses data sets.

The following table lists the data sets that the migrate function creates and uses.

Table 27. Work data sets for MIG

Default data set name	Description	Template keyword
prefix.worklist.DDL	DDL and DML that is constructed from the catalog	MISQL

Table 27. Work data sets for MIG (continued)

Default data set name	Description	Template keyword
prefix.worklist.DDDL	DROP statements for drop objects	MISDROP
prefix.worklist.COL	Identity column information	MICOL
prefix.worklist.CMD	Rebind output	MIGCMD
prefix.worklist.MIGVARS	Partitioned data set for ISPF tables that are required for generating the MIG jobs in batch	MIGSHVR
prefix.worklist.ADB28W1U	Work statement list (WSL) data set	MIUCONV
prefix.worklist.ADB28W3U	WSL data set	MIUOTHR
prefix.worklist.ADB28WDD	WSL elements	MI2WDD
prefix.worklist.ADB28W2T	Input data set for the merge program	MIMLSIN
prefix.worklist.ADB28W2U	Intermediate data set used by the merge program	MIMLSOT

The migrate function also uses data sets for the unloaded data, LOAD control statements, and converted LOAD control statements. The naming convention for the data sets differ depending on whether the Db2 UNLOAD utility or Db2 High Performance Unload is used to unload the data.

The following table lists the data sets for migrations with Db2 UNLOAD.

Table 28. Work data sets for MIG with Db2 UNLOAD

Default data set name	Description	Template keyword
prefix.worklist.CNT.S*n*	LOAD utility control statements, where S*n* is a string assigned to the object by Db2 Admin Tool, with *n* beginning with 1	PUNCHDDN[1]
prefix.worklist.ULD	Data sets for unloaded data	UNLDDN[2]
prefix.worklist.CNC.S*n*	Converted LOAD utility control statements, where S*n* is a string assigned to the object by Db2 Admin Tool, with *n* beginning with 1	MICTLOV (for table spaces MICTLOU (for tables)

Note:

1. A utility template. A template statement is not generated in the JCL. Db2 Admin Tool uses the utility template to generate regular JCL to perform the unload.
2. A utility template. A template statement is generated in the JCL. When you use your own copy of utility template UNLDDN, Db2 Admin Tool does not delete any of the data sets that are created by the template after they are used. You must delete them. Also, the transfer data set list in jobs SST4XF and xxxxSE do not include the data set names, and you must transfer them.

Image copy uses the regular utility template.

The following table lists the data sets for migration with HPU.

Table 29. Work data sets for MIG with HPU

Default data set name	Description	Template keyword
prefix.worklist.CNT.T*n*	LOAD utility control statements, where T*n* is a string assigned to the object by Db2 Admin Tool, with *n* beginning with 1	MICTLIU
prefix.worklist.ULD.T*n*	Unload data sets for a non-partitioned object, where T*n* is a string assigned to the object by Db2 Admin Tool, with *n* beginning with 1	MIDTVNP
prefix.worklist.ULD.T*n*.P*m*	Unload data sets for a partitioned object, where T*n* is a string assigned to the object by Db2 Admin Tool, with *n* beginning with 1, and P*m* is a string that identifies the object's partition number, with *m* beginning with 0001	MIDATVP
prefix.worklist.CNC.T*n*	Converted LOAD utility control statements, where T*n* is a string assigned to the object by Db2 Admin Tool, with *n* beginning with 1	MICTLOU

The relationship between the table name and T*n* and the relationship between the table space name and the S*n* are listed as comments in the beginning of the generated job or WSL.

The migrate function deletes these data sets when they are no longer needed.

Defining naming conventions for MIG work data sets

You can use templates to create your own naming conventions for the work data sets that are created by the migrate (MIG) function.

Procedure

To create naming conventions for the work data sets that are created by the migrate function:

1. On the **Migrate Parameters (ADB28M)** panel, specify YES in the **Generate template statements** field.
2. Specify the TU primary command to manage the templates for the work data sets.

   (Alternatively, you can specify Option 5 on the **DB2 Administration Menu (ADB2)** panel.)

   Use the following variables to specify the data set name pattern for a migrate work data set template:

   Functional variables:

   &ADB28PRE.
   Prefix for data sets specified on the **Migrate Parameters (ADB28M)** panel

   &DB2SYS.
   The Db2 subsystem ID.

   &WORKLIST.
   The work list name that is specified on the **Migrate Parameters (ADB28M)** panel.

   Variables for Db2 utility template processing:
   The following list is a subset of the variables that are supported for normal Db2 utility template processing. The other variables that are supported for Db2 utility processing are not valid for migrate work data sets.

   &DB.
   Database name

   &TS.
   Table space name

&PART.

The value is ALL with these exceptions:

- For template UNLDDN, Db2 resolves the variable to a 5-byte string (*nnnnn*) that represents the partition number. For a non-partitioned object, the value of the string is '00000'. For a partitioned object, the value of the string is '00001', '00002', and so on.
- For template MIDATVP with parallel processing specified, Db2 Admin Tool resolves the variable to a 4-byte string (*nnnn*) that represents the partition number. The value of the string is '0001', '0002', and so on.

&USERID.

Batch user ID

&DATE.

YYYYDD

&TIME.

HHMMSS

&JDATE.

YYYYDDD

&YEAR.

YYYY

&MONTH.

MM

&DAY.

DD

&JDAY.

DDD

&HOUR.

HH

&MINUTE.

MM

&SECOND.

SS

Restriction: The following restrictions apply when specifying variables:

- For the data set names to which Db2 Admin Tool appends S*n*, T*n* or T*n*.P*mmmm* , the number *n* starts with 1 and ends with the number of objects that you want to migrate. The total length of a data set name should not exceed 44 bytes
- The only variables that can be specified for UNLDDN (used when using Db2 UNLOAD) are &DB., &TS., &USERID., and &PART..

  A work list name can be a very important part of the data set name when migrating objects. To specify a work list name as part of UNLDDN template data set name to maintain a consistent naming convention with other data set names, explicitly specify a value in the template instead of using the variable &WORKLIST, which cannot be specified.
- The only variables that can be specified for MIDTVNP and MIDATVP (used when using HPU) are &DB., &TS., &USERID., &WORKLIST., and &PART.. If you specify &PART. for MIDATVP, you must specify it as the last part of the name (for example, &USERID..&TS..ULDULD.P&PART.); otherwise, parallelism will not be performed.

Implicit LOB and XML table support

The Db2 Admin Tool ALT and MIG functions and IBM Db2 Object Comparison Tool for z/OS support changes to implicit LOB and XML table spaces.

The Db2 Admin Tool ALT, and MIG functions and IBM Db2 Object Comparison Tool for z/OS generate multiple image copies when there are implicit LOB or XML table spaces defined for the tables. Generating multiple image copies requires that either a SYSCOPY TEMPLATE is defined and used for the operations, or that the default is used. If no template is provided, this default is used:

```
DSN(&US..&SSID..&DB..&SN..&UQ)
```

The Db2 Admin Tool ALT and MIG functions and IBM Db2 Object Comparison Tool for z/OS generate multiple image copies when there are implicit LOB or XML table spaces defined for the tables. Generating multiple image copies requires that either a SYSCOPY TEMPLATE is defined and used for the operations, or that the default is used. If no template is provided, this default is used:

```
DSN(&US..&SSID..&DB..&SN..&UQ)
```

For clones, this default is used:

```
DSN(&US..&SSID..&DB..&SN..CLONE.&UQ)
```

Work statement lists (WSLs)

A *work statement list*, or WSL, is a collection of one or more tasks that perform basic operations, such as creating objects, granting authorizations, and running utilities.

You can use WSLs to group and run related tasks. For example, suppose that you want to run a batch program that makes a lot of updates, schedule an image copy, and then run the RUNSTATS utility immediately afterward. In this case, you can create a WSL that includes these three tasks. You can then run the WSL online or in batch mode. If the WSL fails, you can restart it.

Although you specify which operations to include in a WSL, the WSL format is created and maintained by Db2 Admin Tool. In general, the statements in a WSL are standard statements or commands that you would normally code to perform a task.

A WSL can also be cloned with different owners and names for the objects. This capability allows you to create a WSL on one subsystem and then change or execute it on another system. For example, the following scenarios are possible:

- **Local use only:** Generate the WSL on subsystem DB2-1. Clone the WSL many times with different owners and names for the objects. Execute the WSL on DB2-1.
- **Local customization and remote execution:** Generate the WSL on subsystem DB2-1. Clone the WSL many times with different owners and names for the objects. Send the WSL to subsystem DB2-2. Execute the WSL on DB2-2.
- **Remote customization and execution:** Generate the WSL on subsystem DB2-1. Send the WSL to DB2-2. Clone the WSL many times with different owners and names for the objects. Execute the WSL on DB2-2.

WSL format

Work statement lists (WSLs) are stored in ISPF tables in a data set that you specify. WSLs can be accessed by other users and are protected by RACF. By storing WSLs in ISPF tables, they can easily be moved to other systems or installations.

```
+---------------------NOT programming interface information----------------------+
```

Each row in the ISPF table contains a work statement, and the order of these rows is important.

Note: WSLs are intended to be created and maintained by Db2 Admin Tool, and the implementation is subject to change. Therefore, although you can manually edit a work statement list (WSL), doing so can produce unexpected results.

A WSL can include the following items. These items are passed as input to ADBTEP2.

SQL statements

The following SQL statements are supported:

- Data definitions, such as CREATE, DROP, ALTER, and RENAME
- Authorization changes, such as GRANT and REVOKE
- Data manipulation changes, such as INSERT, UPDATE, and DELETE

Db2 commands

The format is -*command*.

Example: `-DIS GROUP`

DSN Commands

The following DSN commands are supported:

- BIND
- DCLGEN
- FREE
- REBIND
- RUN

Db2 Utility statements

The following Db2 utilities are supported:

- CHECK
- COPY
- COPYTOCOPY
- DIAGNOSE
- LOAD
- MERGECOPY
- MODIFY
- QUIESCE
- REBUILD
- RECOVER
- REORG
- REPAIR
- REPORT
- RUNSTATS
- STOSPACE
- UNLOAD

UTILFROM statements

The Db2 UTILFROM utility is a pseudo utility that directs ADBTEP2 to execute the utility control statements that are contained in a data set. Only one utility can be contained within the data set. Therefore, RUNSTATS and LOAD cannot both be included in one UTILFROM statement. The purpose of this utility is to allow a ADBTEP2 to execute the LOAD control statements that are generated by UNLOAD, REORG UNLOAD, and Db2 HPU. Because UNLOAD does not produce all the options that are required (such as SORTNUM), they can be added by using the ADD keyword.

Use the following format of UTILFROM with the ADD parameter:

```
UTILFROM dsname ADD(additional options)
```

The following example shows the ADD parameter:

```
UTILFROM ROYC.ROYCDOC1.CNTLC.PPP2
   ADD(SORTNUM 8 SORTDEVT SYSDA
     WORKDDN(UTLUT1,UTLOUT) ERRDDN(UTLERR)
     DISCARDDN(UTLDISC) MAPDDN(UTLMAP));
```

To add DISCARDDN information at the partition level during a LOAD operation, use the ADDPART parameter, as shown in the following example:

```
UTILFROM TS5794.DC1Q.C0000602.CNT.T0001
ADDPART(DISCARDDN(EJDISC))
ADD(SORTNUM 8 SORTDEVT SYSDA
WORKDDN(UTLUT1,UTLOUT) ERRDDN(UTLERR)
MAPDDN(UTLMAP) ENFORCE NO DISCARDS 0);
```

Functional comments

The following functional comments are supported:

--#SET ROWS_FETCH *n*

where *n* is a non-negative integer that indicates the maximum number of rows to be fetched for each subsequent SELECT statement. Use -1 to indicate that all rows should be fetched.

--#SET ROWS_OUT *n*

where *n* is a non-negative integer that indicates the maximum number of rows to be output for each subsequent SELECT statement. Use -1 to indicate that all rows should be output.

--#SET TERMINATOR *n*

where *n* is a one-byte character that is to terminate the next SQL statement. Any character is valid, except blank, comma, single quotation, double quotation, underscore, and parentheses.

--#SET ACCEPT_RC (ON/OFF) *m n*

where *m*or *n* is the SQLCODE that can be accepted for the SQL statements before the program stops. The maximum number of SQLCODE that can be listed is 5. Using --#SET ACCEPT_RC *m n* can accept SQLCODE *m* or *n* for the following single SQL statement. Using --#SET ACCEPT_RC ON *m n* can accept SQLCODE *m* or *n* for the following multiple SQL statements until the next --#SET ACCEPT_RC OFF occurs. If no SQLCODE is provided after --#SET ACCEPT_RC (ON/OFF), it means all SQLCODEs can be accepted.

--#SET MAXERRORS *n*

where *n* is the number of DSN commands that can fail before the program stops. Use -1 to indicate that the program should tolerate an unlimited number of errors for DSN commands.

--#PROCESS *name*

where *name* is the name of a process, such as DROP or UNLOAD. This comment indicates that the WSL is split and these processes are to be run as parallel jobs.

Note: IBM reserves the right to use additional parameters in these functional comment statements. These parameters might be present in the statements that Db2 Admin Tool generates for ADBTEP2. Do not modify these statements unless you are requested to do so by your IBM service representative.

REXX execs

The format is *`REXX execname parameters`*

`execname` can be the name of a CLIST. Programs are not supported. Db2 programs can be executed by using the DSN command RUN.

Db2 Admin Tool instructions

These commands follow a product-specific syntax for performing certain complex operations. The syntax of these instructions is subject to change.

WSLs can include the following Db2 Admin Tool commands:

RESTART

Defines a restart point for the WSL.

If you run a WSL and it fails before completing, you can restart it from any defined restart point or from the point where it failed. A WSL can have many restart points, but only one point can be used for the restart.

The syntax is

```
--#RESTART name
```

where *name* is the name of the restart point. This name can be anything except YES, NO, FORCE, or a pure numeric value.

ADM PARALLEL and ADM ENDPARALLEL
ADM JOB and ADM ENDJOB

Indicates that statements are to run in parallel.

Some of the input processes to the WSL (for example, from Object Comparison Tool) include this parallel specification.

The ADM PARALLEL and ADM ENDPARALLEL statements signify the start and end points for jobs to be run in parallel. Within the PARALLEL and ENPARALLEL pair, the ADM JOB and ADM ENDJOB statements signify the start and end points for a particular job.

Optionally, a name can be specified for the parallel process in the following format:

```
ADM PARALLEL name
```

WSL statements that are not included in a PARALLEL and ENDPARALLEL pair are placed in a separate job.

For example, the following specification results in three jobs. The first two jobs run concurrently and the third one runs when the first two are complete.

```
ADM PARALLEL UNLOAD

ADM JOB
tasks for job1
ADM ENDJOB

ADM JOB
tasks for job2
ADM ENDJOB

ADM ENDPARALLEL

serial tasks
```

A parallel specification might be used to unload multiple tables in parallel, then run DDL to drop and redefine the tables, and then load the tables in parallel. In this case, the loads and unloads can be run in parallel to increase performance. The DDL is done in one job to avoid Db2 locking or serialization problems.

ADMIN ACCELERATOR

Performs some function on the accelerator.

Examples:

ADMIN ACCELERATOR ADD

Defines a table on the accelerator.

ADMIN ACCELERATOR ARCHIVE

Archives parts of partitioned by range table or tables on the accelerator or accelerators.

ADMIN ACCELERATOR DELETE

Deletes a table from the accelerator.

ADMIN ACCELERATOR DISABLE

Disables acceleration for the table or tables defined on the accelerator or accelerators.

ADMIN ACCELERATOR ENABLE

Enables acceleration or replication for the table or tables defined on the accelerator or accelerators.

ADMIN ACCELERATOR LOAD

Loads data of the accelerated table or tables to the accelerator.

ADMIN IF and ADMIN ENDIF

Defines a conditional block of statements, which may or may not be processed.

ADMIN IF DEFINEYES

Checks whether a table space has the DEFINE YES attribute. Some utilities do not work if the table space has the DEFINE NO attribute.

ADMIN IF DSEXISTS

Checks whether UNLOAD created a LOAD statement.

ADMIN REGISTER TOKEN

Registers whether UNLOAD was run.

Db2 Admin Tool support commands

These commands are associated with (or support) primary commands that are listed later in the batch statement list. For example, the ALLOC command is used to allocate files for a program (the primary command). Support command processing is deferred until the primary command is encountered. Support commands must immediately precede their primary command.

The following commands are considered Db2 Admin Tool support commands:

ADBSYSIN

Identifies the location of the input.

Many programs, including ADBTEP2, use the filename (or DDNAME) SYSIN. ADBTEP2 uses SYSIN for the batch statement list; therefore, ADBSYSIN is used to identify the location of the input. The format is

```
ADBSYSIN COPYDD(ddname)
```

where *ddname* contains the SYSIN for the program following the ADBSYSIN.

ADBPAUSE

Pauses the current run of ADBTEP2 or ADBTEPA at a certain point. ADBTEP2 and ADBTEPA can be restarted at that point.

ALLOC

Issues a TSO ALLOCATE command with the provided parameters. ALLOC is intended to support programs only. It is not a valid support command for a Db2 utility.

Example: `ALLOC DD(DATAI001) DS('ROYC.ROYCDOC1.UNLD.PPP1') SHR`

CHECKBEGIN and CHECKEND

Delimits a block of CHECK DATA statements. When CHECKEND is reached, Db2 Admin Tool identifies the parent and child tables in RI relationships with the table spaces that are identified in the CHECK DATA statements within the block and generates CHECK DATA statements to clear these tables of any CHECK-pending status. Any TSODELETE commands before the CHECKEND statement are executed for all the generated CHECK DATA statements. Any TEMPLATE statements before the CHECKEND are supplied to the utility for all the generated CHECK DATA commands.

Example: In the following example, the second set of TSODELETE and TEMPLATE statements apply to the CHECK DATA statements that might be generated for the parent and descendent tables:

```
CHECKBEGIN;
TSODELETE 'JIMWC.EB12.CSUT1.T0001';
TSODELETE 'JIMWC.EB12.CSOUT.T0001';
TSODELETE 'JIMWC.EB12.CSERR.T0001';
TEMPLATE UTLUT1 DSN 'JIMWC.EB12.CSUT1.T0001'
         UNIT SYSDA;
TEMPLATE UTLOUT DSN 'JIMWC.EB12.CSOUT.T0001'
         UNIT SYSDA;
TEMPLATE UTLERR DSN 'JIMWC.EB12.CSERR.T0001'
         UNIT SYSDA;
CHECK DATA TABLESPACE DB2144.TS2144
  ERRDDN(UTLERR) WORKDDN(UTLUT1,UTLOUT)
  SORTDEVT SYSDA SORTNUM 4;
TSODELETE 'JIMWC.EB12.CSUT1.T0001';
TSODELETE 'JIMWC.EB12.CSOUT.T0001';
TSODELETE 'JIMWC.EB12.CSERR.T0001';
TEMPLATE UTLUT1 DSN 'JIMWC.EB12.CSUT1.T0001'
         UNIT SYSDA;
TEMPLATE UTLOUT DSN 'JIMWC.EB12.CSOUT.T0001'
         UNIT SYSDA;
TEMPLATE UTLERR DSN 'JIMWC.EB12.CSERR.T0001'
         UNIT SYSDA;
CHECKEND;
```

TEMPLATE

Defines a template to support utilities. ADBTEP2 passes this command to the Db2 utility processor. ADBTEP2 performs a partial simulation of the Db2 TEMPLATE function for template names that are not supported by Db2 (for example, SYSREC). The main difference between Db2 allocation of templates and the simulation is at failure, as the failure disposition is not honored. ADBTEP2 does not support utility wild cards.

TSODELETE

Issues a TSO DELETE command for the data set provided. If the DELETE fails, a DELETE NOSCRATCH is attempted. Processing continues even if TSODELETE is unsuccessful.

Delimiters

Each statement must end with the current delimiter character. The delimiter character is a semicolon unless a -#SET TERMINATOR functional comment precedes the statement. (A semicolon is the delimiter for all ADBTEP2 commands, and a WSL calls ADBTEP2.)

Quotation marks can be used with delimited identifiers in a statement. If you clone a WSL that includes a statement containing delimited identifiers, Db2 Admin Tool removes the quotation marks from the identifier if it does not require delimiters.

Example: A WSL contains the following DDL:

```
DDL  CREATE SYNONYM "PROJSYN" FOR "DBA282"."PROJ"
```

The cloned result does not contain the quotation marks:

```
COM  -- Created by DBA282 on 2002/07/23 at 15:23 by cloning of
COM  -- source work stmt list  RESULT  from library  WSL.DATA
DDL  CREATE SYNONYM PROJSYN FOR DBA282.PROJ
```

```
+------------------End of NOT programming interface information------------------+
```

Related tasks

“Converting the WSL format in CM batch” on page 559

If you use Change Management (CM) batch, you can convert a work statement list (WSL) between internal ISPF table format and readable format. This conversion allows you to view and potentially edit a WSL without using Db2 Admin Tool panels.

Related information

“Programming interface information” on page 85

Creating a WSL

Work statement lists (WSLs) are created during the process of performing some other task in Db2 Admin Tool. For example, when creating an object, you can choose to add the CREATE statement to a WSL. At that point, you can create a new WSL. Alternatively, you can append to or replace an existing WSL.

The process of creating a WSL (or adding an operation to an existing WSL) varies slightly depending on what task you are performing, as described in the following information:

- "Adding SQL and commands to a WSL" on page 537
- "Directing the output of the GEN command to a WSL" on page 538
- "Directing the output of the DDL line command to a WSL" on page 538
- "Adding alter (ALT) operations to a WSL" on page 539
- "Adding apply jobs to a WSL" on page 539
- "Adding Db2 utility statements to a WSL" on page 540

Alternatively, you can create a new WSL by cloning an existing WSL. See "Cloning a WSL" on page 543.

Adding SQL and commands to a WSL

About this task

You can add the following statements and commands to a WSL:

- Definition SQL (CREATE, DROP, ALTER, and RENAME)
- Authorization SQL (GRANT and REVOKE)
- Update SQL (INSERT, UPDATE, and DELETE)
- DSN commands (BIND, REBIND, FREE, and RUN)
- Db2 commands (START, STOP, ALTER, and SET)
- REXX and CLIST statements

Procedure

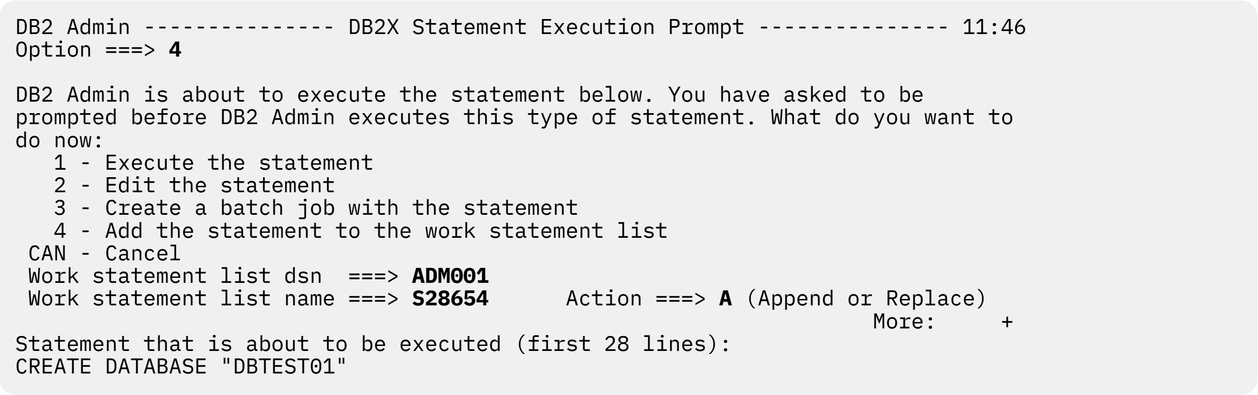
To add SQL and commands to a WSL:

1. Enable the use of WSLs for these operations by activating prompting:
   a) Issue the PROMPT primary command.
   b) On the **Prompt Options (ADB2PRMT)** panel, specify YES next to any operation type for which you want to allow the use of WSLs.

      For more information about this panel and prompt options, see "Changing Db2 Admin Tool prompt options" on page 243.

      REXX and CLIST statements are not activated by using PROMPT. No comparable method exists.

   When you later run the selected operations, the **Statement Execution Prompt (ADB2PSTM)** panel is displayed.
2. On the **Statement Execution Prompt (ADB2PSTM)** panel, select option 4, specify the appropriate WSL information, and press Enter.



```
DB2 Admin --------------- DB2X Statement Execution Prompt --------------- 11:46
Option ===> 4

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
   1 - Execute the statement
   2 - Edit the statement
   3 - Create a batch job with the statement
   4 - Add the statement to the work statement list
 CAN - Cancel
 Work statement list dsn  ===> ADM001
 Work statement list name ===> S28654      Action ===> A (Append or Replace)
                                                                  More:     +
Statement that is about to be executed (first 28 lines):
CREATE DATABASE "DBTEST01"
```

*Figure 290. **Statement Execution Prompt (ADB2PSTM)** panel*

To create a new WSL, specify the following information:

- In the **Work statement list name** field, specify a new name.
- In the **Work statement list dsn** field, specify the name of a data set to store the WSL. If the data set exists, it must be defined with RECFM=FB, LRECL=80 and DSORG=PO (a PDS). If it does not exist, it will be created.

To add this operation to an existing WSL, specify the following information:

- In the **Work statement list name** field, specify the name of the existing WSL.
- In the **Work statement list dsn** field, specify the name of the data set where that WSL is stored.
- In the **Action** field, specify whether you want to append this operation to that WSL or replace that WSL with this operation.

Directing the output of the GEN command to a WSL

About this task

You can add the SQL that is output from the reverse engineering function (the GEN command) to a WSL.

As part of the task of "Generating SQL to re-create a Db2 object (reverse engineering)" on page 357, the **Generate SQL from DB2 catalog (ADB2GENB)** panel is displayed.

Procedure

To direct the output of the GEN command to a WSL:

1. On the **Generate SQL from DB2 catalog (ADB2GENB)** panel, in the **Add to work stmt list** field, specify Y.
2. On the **Specify Work Statement List (ADB2WLDA)** panel, specify the WSL data set and member name.

   If the WSL already exists, in the **Existing name action** field, specify whether to append the GEN output to the end of the current contents of that WSL or to replace the current contents of the WSL with the GEN output.

Directing the output of the DDL line command to a WSL

Procedure

To direct the output of the DDL line command to a WSL:

1. Issue the command `PROMPT ON`.

   This command activates prompting. It has the same effect as the steps that are described in Step 1 (Adding SQL and commands to a WSL).

2. Issue the DDL line command against an object.

   This line command can be issued from multiple panels. For example, you can issue the DDL line command from object panels that you navigate to from the **System Catalog (ADB21)** panel.
3. Issue the EXECUTE command.
4. On the **Statement Execution Prompt (ADB2PSTM)** panel, select option 4, specify the appropriate WSL information, and press Enter.

Adding alter (ALT) operations to a WSL

About this task

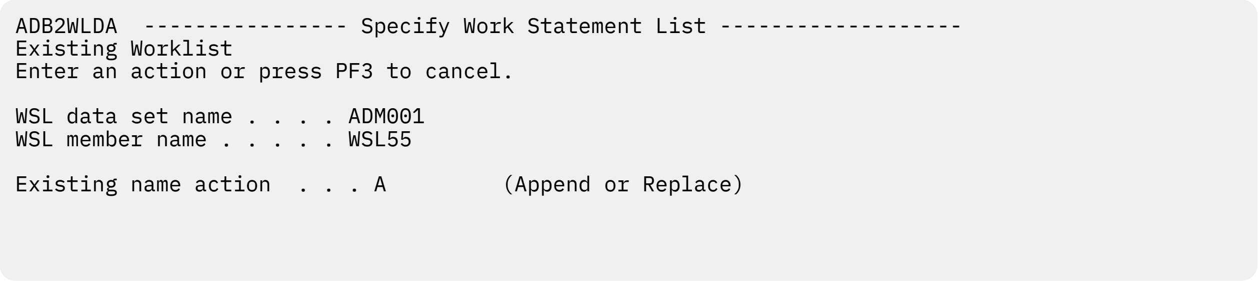
You can alter multiple tables at the same time by appending several ALTER TABLE (ALT) requests to one WSL.

As part of the task of altering objects, the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel is displayed so you can specify options for the alter operation.

Procedure

To add alter (ALT) operations to a WSL:

1. On the **ALTER - Build Analyze and Apply Job (ADBPALT)** panel, in the **As work statement list** field, specify YES.
2. On the **Specify Work Statement List (ADB2WLDA)** panel, specify the WSL information and press Enter:



```
ADB2WLDA  ---------------- Specify Work Statement List -------------------
Existing Worklist
Enter an action or press PF3 to cancel.

WSL data set name . . . . ADM001
WSL member name . . . . . WSL55

Existing name action  . . . A         (Append or Replace)
```

   Figure 291. ***Specify Work Statement List (ADB2WLDA)*** *panel*

   You must specify the library data set name and member name. If the WSL name already exists, you can choose to add the ALT output to the end of the current contents of that WSL or to replace the current contents of the WSL with the ALT output.

   The next panel displays the JCL that you must run to populate the WSL.
3. Edit the JCL as needed and submit the job.

Adding apply jobs to a WSL

About this task

If you use Db2 Object Comparison Tool, you can add the generated apply jobs to a WSL. (These jobs apply to the target any changes that were found during the comparison.)

As part of the comparison process, the **Generate Compare Jobs (GOC5)** panel is displayed so you can specify options for the comparison. This panel is displayed when you select option 5 on the **DB2 Object Comparison Tool Menu (GOCMENU)** panel.

Procedure

To add apply jobs to a WSL:

1. On the **Generate Compare Jobs (GOC5)** panel, in the **As work statement list** field, specify YES.
2. When prompted for WSL information by the **Specify Work Statement List (ADB2WLDA)** panel, specify the appropriate information and press Enter.

   Specify the WSL library data set name and member name. If the WSL name already exists, you can choose to add the apply jobs to the end of the current contents of that WSL or to replace the current contents of the WSL with the apply jobs.

   The next panel displays the JCL that you must run to populate the WSL.
3. Edit the JCL as needed and submit the job.

Adding Db2 utility statements to a WSL

About this task

You can add output from storage group, table space, table, and index utilities to a WSL

Procedure

To add utility statements to a WSL:

1. In the process of requesting that a Db2 utility be run, you can specify that the utility statements be added to a WSL on any of the following panels:
   - **Tables Utilities (ADB2UT)** panel
   - **Table Space Utilities (ADB2US)** panel
   - **Index Utilities (ADB2UX)** panel
   - **Storage Group Utilities (ADB2UG)** panel
   - **Create Index Utilities (ADB26CXU)** panel
   - **LISTDEF Utilities (ADB25LU)** panel
2. When prompted, specify the WSL library and member name. If the WSL name already exists, you can choose to add the utility statements to the end of the current contents of that WSL or to replace the current contents of the WSL with the utility statements.

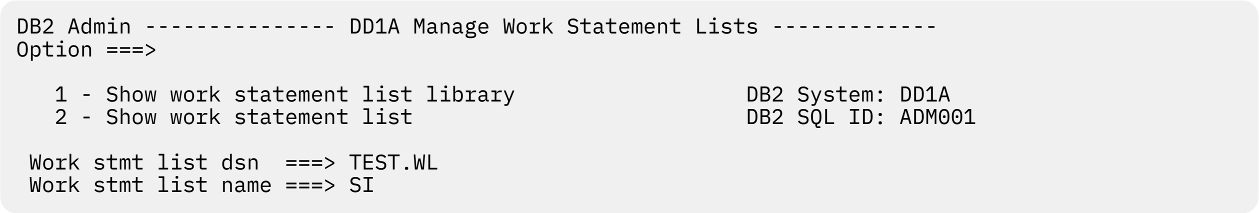
Viewing a WSL library

A *work statement list (WSL) library* contains one or more work statement lists (WSLs). You can edit or delete any WSLs in the library.

Procedure

To view a WSL library:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option `W`, and press Enter.
2. On the **Manage Work Statement Lists (ADB2W)** panel, in the **Work stmt list dsn** field, specify the data set name of the ISPF library that contains the WSLs.

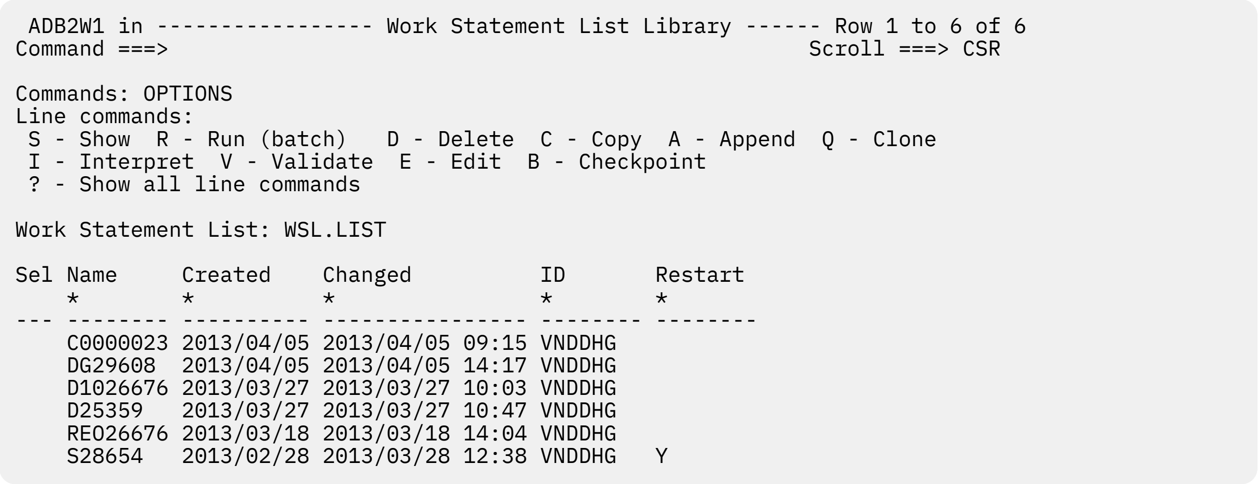


```
DB2 Admin --------------- DD1A Manage Work Statement Lists -------------
Option ===>

   1 - Show work statement list library                  DB2 System: DD1A
   2 - Show work statement list                          DB2 SQL ID: ADM001

 Work stmt list dsn  ===> TEST.WL
 Work stmt list name ===> SI
```

*Figure 292. **Manage Work Statement Lists (ADB2W)** panel*

3. Specify option 1, and press Enter.

   The **Work Statement List Library (ADB2W1)** panel shows the contents of the WSL library. Each WSL is listed on a separate line.



```
 ADB2W1 in ----------------- Work Statement List Library ------ Row 1 to 6 of 6
Command ===>                                                  Scroll ===> CSR

Commands: OPTIONS
Line commands:
 S - Show  R - Run (batch)   D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit  B - Checkpoint
 ? - Show all line commands

Work Statement List: WSL.LIST

Sel Name     Created    Changed          ID       Restart
    *        *          *                *        *
--- -------- ---------- ---------------- -------- --------
    C0000023 2013/04/05 2013/04/05 09:15 VNDDHG
    DG29608  2013/04/05 2013/04/05 14:17 VNDDHG
    D1026676 2013/03/27 2013/03/27 10:03 VNDDHG
    D25359   2013/03/27 2013/03/27 10:47 VNDDHG
    RE026676 2013/03/18 2013/03/18 14:04 VNDDHG
    S28654   2013/02/28 2013/03/28 12:38 VNDDHG   Y
```

Figure 293. Work Statement List Library panel (ADB2W1)

From this panel you can perform actions on each WSL. For example, you can view or run a WSL. Use the following line commands to perform actions as needed:

S

Shows the WSL contents.

R

Runs the WSL. For more information about running a WSL online or in batch, see "Running a WSL" on page 551.

D

Deletes the WSL from the library.

C

Copies the WSL and appends it to the WSL that is indicated by the A line command.

A

Appends to this member the WSL that is indicated by the C line command.

Q

Clones the WSL member for use on other Db2 subsystems. For more information about cloning a WSL, see "Cloning a WSL" on page 543.

I

Interprets the WSL. For more information about interpreting a WSL, see "Interpreting a WSL" on page 550.

V

Validates the syntax of the SQL statements in the WSL and provides an impact analysis of the objects that would be affected by running the WSL. For more information about validating a WSL, see "Validating WSLs" on page 546.

E

Invokes ISPF EDIT so you can edit the WSL. Upon exiting from EDIT mode, the original WSL is updated.

Note: WSLs are intended to be created and maintained by Db2 Admin Tool, and the implementation is subject to change. Therefore, although you can manually edit a work statement list (WSL), doing so can produce unexpected results.

Tip: To perform a search for a string in the WSL, invoke the EDIT command to display all of the statements, then use FIND to search for a specific text string.

For information about the format of a WSL, see "WSL format" on page 531.

Viewing a WSL

You can view the content of a single work statement list (WSL). Optionally, you can then edit the order (sequence) of the statements in the WSL.

Procedure

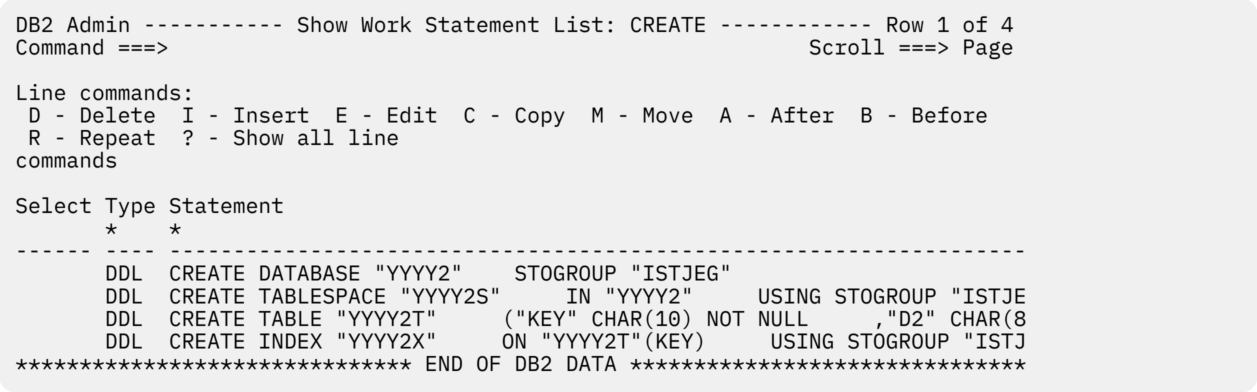
To view a WSL:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option W, and press Enter.
2. On the **Manage Work Statement Lists (ADB2W)** panel, specify the following information:

   Work stmt list dsn
   The data set name of the ISPF library that contains the WSL.

   Work stmt list name
   The name of the WSL. If the WSL does not exist, Db2 Admin Tool will create it for you.
3. Specify option 2, and press Enter.

   The WSL is displayed:



```
DB2 Admin ----------- Show Work Statement List: CREATE ------------ Row 1 of 4
Command ===>                                                  Scroll ===> Page

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat  ? - Show all line
commands

Select Type Statement
       *    *
------ ---- --------------------------------------------------------------------
       DDL  CREATE DATABASE "YYYY2"    STOGROUP "ISTJEG"
       DDL  CREATE TABLESPACE "YYYY2S"     IN "YYYY2"     USING STOGROUP "ISTJE
       DDL  CREATE TABLE "YYYY2T"     ("KEY" CHAR(10) NOT NULL     ,"D2" CHAR(8
       DDL  CREATE INDEX "YYYY2X"     ON "YYYY2T"(KEY)     USING STOGROUP "ISTJ
******************************* END OF DB2 DATA ********************************
```

Figure 294. ***Show Work Statement List (ADB2W1S)*** *panel*

Each statement in the WSL is listed on a separate line. The **Type** column indicates the type of statement and can have one of the following values:

ADM
A Db2 Admin Tool statement

COM
A comment statement

DB2
A Db2 START, STOP, or SET command

DCL
An authorization change SQL statement, such as GRANT and REVOKE

DDL
A data definition SQL statement, such as CREATE, ALTER, and DROP

DML
A data manipulation SQL statement, such as INSERT, UPDATE and DELETE

DSN
A DSN BIND, REBIND, or FREE command

UTL
A Db2 utility statement

You use the following line commands to modify the WSL:

D
Delete the statement from the list.

I
: Insert a statement into the list.

E
: Edit the statement.

  Note: WSLs are intended to be created and maintained by Db2 Admin Tool, and the implementation is subject to change. Therefore, although you can manually edit a work statement list (WSL), doing so can produce unexpected results.

C
: Copy this statement to the line identified by an A (after) or a B (before) line command.

M
: Move this statement to the line identified by an A (after) or a B (before) line command.

A
: Identifies that the destination of a move or copy operation is after this line.

B
: Identifies that the destination of a move or copy operation is before this line.

R
: Repeat the statement

You can issue the C and M line commands in a separate operation from the A and B line commands. If entered separately, the first line command encountered remains pending until its counterpart is encountered. While a line command is pending, any intervening line commands (such as E for edit) can be processed. However, if a line is deleted while in pending state, the operation is removed.

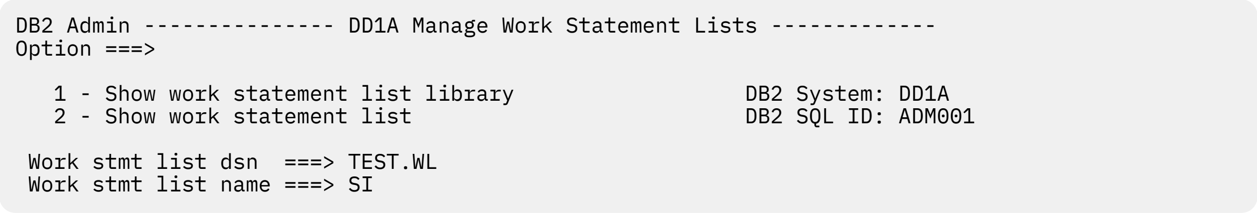
Cloning a WSL

You can clone a work statement list (WSL) to use on another Db2 subsystem or against Db2 objects with different naming schemes. Cloning always leaves the original WSL unchanged.

Procedure

To clone a WSL:

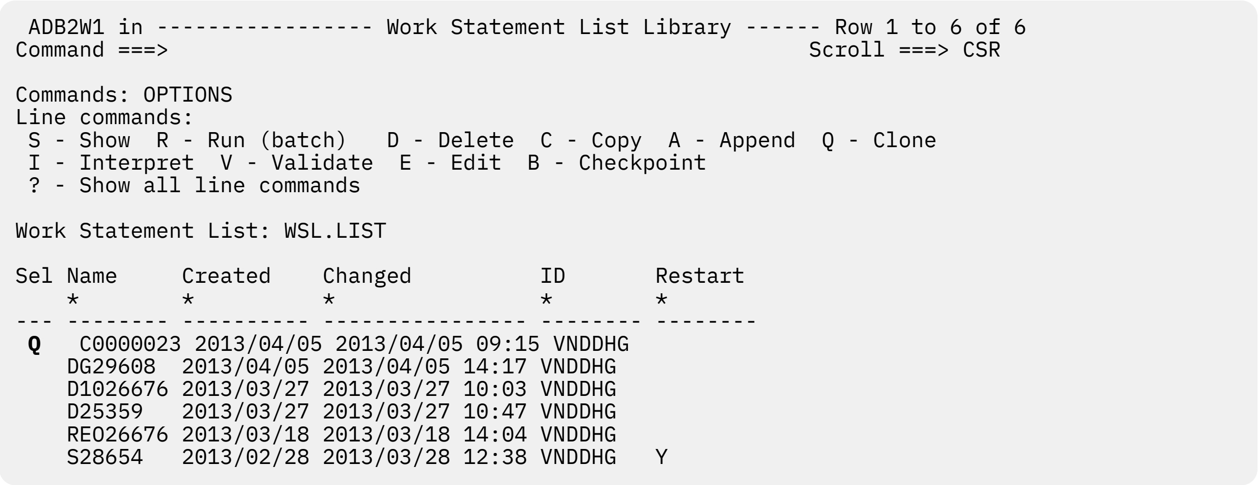
1. On the **DB2 Administration Menu (ADB2)** panel, specify option W, and press Enter.
2. On the **Manage Work Statement Lists (ADB2W)** panel, in the **Work stmt list dsn** field, specify the data set name of the ISPF library that contains the WSL.



```
DB2 Admin --------------- DD1A Manage Work Statement Lists -------------
Option ===>

   1 - Show work statement list library                 DB2 System: DD1A
   2 - Show work statement list                         DB2 SQL ID: ADM001

 Work stmt list dsn  ===> TEST.WL
 Work stmt list name ===> SI
```

*Figure 295. **Manage Work Statement Lists (ADB2W)** panel*

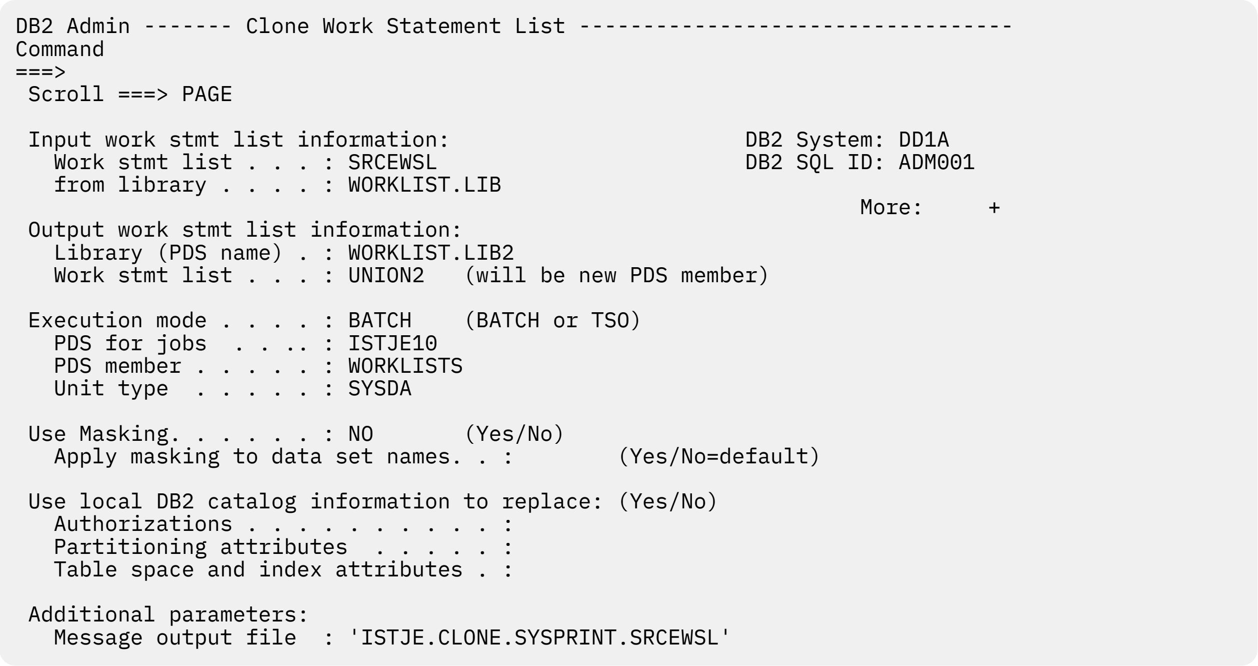
3. Specify option 1, and press Enter.
4. On the **Work Statement List Library (ADB2W1)** panel, specify the Q line command against the WSL that you want to clone, and press Enter:



```
 ADB2W1 in ----------------- Work Statement List Library ------ Row 1 to 6 of 6
Command ===>                                                  Scroll ===> CSR

Commands: OPTIONS
Line commands:
 S - Show  R - Run (batch)   D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit  B - Checkpoint
 ? - Show all line commands

Work Statement List: WSL.LIST

Sel Name     Created    Changed          ID       Restart
    *        *          *                *        *
--- -------- ---------- ---------------- -------- --------
 Q   C0000023 2013/04/05 2013/04/05 09:15 VNDDHG
    DG29608  2013/04/05 2013/04/05 14:17 VNDDHG
    D1026676 2013/03/27 2013/03/27 10:03 VNDDHG
    D25359   2013/03/27 2013/03/27 10:47 VNDDHG
    REO26676 2013/03/18 2013/03/18 14:04 VNDDHG
    S28654   2013/02/28 2013/03/28 12:38 VNDDHG   Y
```

Figure 296. Work Statement List Library panel (ADB2W1)

5. On the **Clone Work Statement List (ADB2W1Q)** panel, specify the following information, and press Enter:



```
DB2 Admin ------- Clone Work Statement List ----------------------------------
Command
===>
 Scroll ===> PAGE

 Input work stmt list information:                       DB2 System: DD1A
   Work stmt list . . . : SRCEWSL                        DB2 SQL ID: ADM001
   from library . . . . : WORKLIST.LIB
                                                                  More:     +
 Output work stmt list information:
   Library (PDS name) . : WORKLIST.LIB2
   Work stmt list . . . : UNION2   (will be new PDS member)

 Execution mode . . . . : BATCH    (BATCH or TSO)
   PDS for jobs  . . .. : ISTJE10
   PDS member . . . . . : WORKLISTS
   Unit type  . . . . . : SYSDA

 Use Masking. . . . . . : NO       (Yes/No)
   Apply masking to data set names. . :       (Yes/No=default)

 Use local DB2 catalog information to replace: (Yes/No)
   Authorizations . . . . . . . . . . :
   Partitioning attributes  . . . . . :
   Table space and index attributes . :

 Additional parameters:
   Message output file  : 'ISTJE.CLONE.SYSPRINT.SRCEWSL'
```

Figure 297. Clone Work Statement List panel (ADB2W1Q)

Note: The **Input work statement list information** displays the WSL that you selected and the library in which it is stored.

Library (PDS name)

Specify the name of a library in which to place the new WSL as a new PDS member. Use standard TSO format for this name. If this PDS does not exist, Db2 Admin Tool creates and catalogs this PDS with a default size of 1 cylinder, record length 80, and fixed-block with a block size of 6160.

Work stmt list

Specify the name of the new (cloned) WSL. Db2 Admin Tool creates a new PDS member with this name in the PDS (library) that you specify. If a PDS member by this name already exists in that library, the PDS member is not replaced, and the cloning attempt fails.

Because WSLs are stored as ISPF tables, they are subject to the ISPF restriction that requires currently active tables to have different names, even when the tables are from different libraries. Therefore, the cloned WSL that you create and its source WSL must have different names.

Execution mode

Specify a mode for running the cloning job, either batch or TSO. If you select batch mode, specify values for the **PDS for jobs** and **PDS member** fields. If necessary, change the default unit type.

Use Masking

Specify whether to use masking.

Masking enables you to change the names of the Db2 objects, owners, and schemas that are referenced in the original WSL to different names in the new (cloned) WSL. Masking also enables you to specify overwrite values for several table space and index space attributes. Masking is often useful when the new (cloned) WSL is to be used on a different Db2 subsystem or in a different database.

If you specify YES, the **Specify Masks (ADB2GENM)** panel is displayed, and you can specify the mask to use and edit the mask definition before you begin the cloning process. For instructions on how to specify masks, see "Specifying masks" on page 315.

Apply masking to data set names

Specify whether name masking is to be applied to data set names. Name masking is useful when Db2 Admin Tool generates data set names with qualifiers that are based on database object names. This field affects only the following statements: TSO ALLOCATE, ADM TSODELETE, UTL TEMPLATE, and UTL UTLFROM(admin).

Authorizations

Overrides authorization to objects that are created by the WSL with authorization records (grants) from the local Db2 catalog. Grants to objects that are not created by the WSL are not overridden.

Partitioning attributes

Overrides characteristics of partitioned tables spaces and indexes in the local Db2 catalog. Objects that are not partitioned in the local Db2 catalog are not affected. The list of columns that comprise the index key is not overridden. This index property is always taken from the WSL statement.

Restriction: Certain conditions make it impossible to override partitioning. For example, changing partitioning attributes is unsafe if the list of index columns in the WSL statement is not a strict extension of the list of index columns found in the local Db2 catalog. In this case (for an index on a table), no partitioning attributes are overridden.

Table space and index attributes

For the CREATE TABLESPACE and CREATE INDEX statements, in the newly cloned WSL, you can replace the primary and secondary quantity values specified in these statements with the values from the local Db2 catalog tables (SYSTABLEPART and SYSINDEXPART) where cloning is requested. If the masking feature is used, the masking to change Db2 object names and owners is performed first. Then any overwrite values that are specified for PRIQTY and SECQTY are applied using the new table space or index names.

Message output file

Specify an output message data set for the cloned WSL

To browse this message output file, use the M primary command.

What to do next

If the target Db2 subsystem exists on a remote site, you can use standard TSO services to send the newly cloned WSL to that remote site. Or, you can send the original WSL to that remote site first, and complete the cloning on that remote site.

Validating WSLs

Validating a WSL enables you to generate a report about the syntax and the impact to other objects.

About this task

Before you run a WSL, you might want to have the syntax of the SQL statements checked and assess the impact that running the WSL would have on objects.

When you validate a WSL, Db2 Admin Tool checks the syntax of each SQL statement in isolation from any other SQL statements in the WSL; it ignores any SQL statements that precede the statement currently being checked. Thus, Db2 Admin Tool can generally report all syntactic errors but might miss semantic errors that can result from not being able to see previous statements. For example, if the name of a data type is required in a certain position in the syntax, Db2 Admin Tool does not verify that the name of the data type is either a built-in data type or a user-defined data type that has been previously defined.

Note: For native SQL procedures, even if validation is successful, the object's existence in the body of the native SQL procedure cannot be known at procedure run time (or during procedure call).

The impact analysis portion of the validate report lists the impact to the objects by these categories:

Implicitly dropped objects
Existing objects that are implicitly dropped but not re-created by the WSL.

Explicitly dropped objects
Existing objects that are explicitly dropped but not re-created by the WSL.

Recreated objects
Existing objects that are implicitly or explicitly dropped and re-created by the WSL.

Altered objects
Existing objects that are altered by the WSL.

Created objects
Objects that did not exist and are created by the WSL.

Temporary objects
Objects that did not exist and are created and then dropped by the WSL.

Each affected object is included in only one of these categories.

After the initial validation, you can add custom validation rules for objects from Object Compare, CM Analyze, and CM Batch functions. To perform a custom validation, you must write the REXX exec that is used to perform the validation. You can specify the name of your REXX exec on the Options for Change Functions panel (ADB2PCO). The name of your REXX exec is then stored in your profile shared pool. During the validation process, the REXX exec name will be added to the JCL as the value for the STMTEXIT parameter. The REXX exit cannot connect to Db2, because the connection between Db2 Admin Tool and Db2 is already active.

The REXX exec will be called with the following parameters:

- Statement type
- Object type
- Object qualifier
- Object name

You can decide how to perform the validation based on these parameters.

The custom validation function will continue processing until all validations are complete and all errors are reported. Validation errors are written to a separate DD file called VALOUT and use the data set name prefix that you specify. An example of a data set name is SYSADM.E344.VALOUT. Each error message from the REXX exec consists of a return code followed by a colon and the message string, as shown in the example below.

```
'08:Changes in database not allowed'
```

The return string from the REXX exec is saved to the VALOUT DD in the format shown in the example below.

```
SUPPLEMENTAL VALIDATE WORK STATEMENT LIST REPORT
================================================
Prepared on DSN7 (DB2 Release 720) by NBRON at 2006-07-08 10:48
for NBRON.WLIST.VALIDATE(SAMPLE)

ADB3036E RC=08 An error occurred while processing the ALTER DB statement:
CHANGES IN DATABASE NOT ALLOWED
```

If an object does not require validation, or if there are no errors, the REXX exec returns the code '00'.

Procedure

To validate a WSL:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option W, and press Enter.
2. On the **Manage Work Statement Lists (ADB2W)** panel, in the **Work stmt list dsn** field, specify the data set name of the ISPF library that contains the WSLs.
3. Specify option 1, and press Enter.
4. On the **Work Statement List Library (ADB2W1)** panel, specify the V line command against a WSL, and press Enter:

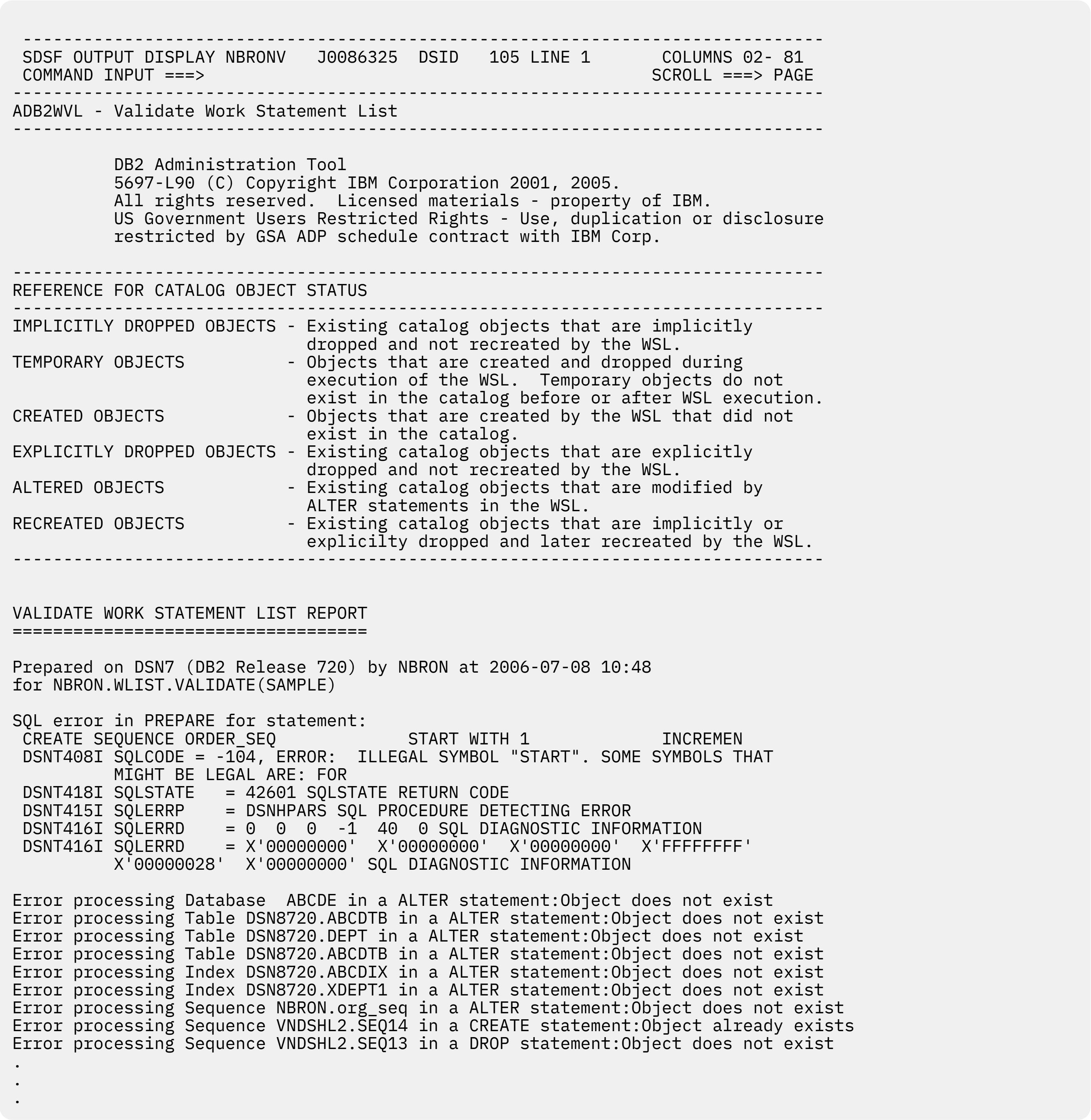
```
 ADB2W1 in ----------------- Work Statement List Library ------ Row 1 to 6 of 6
Command ===>                                                  Scroll ===> CSR

Commands: OPTIONS
Line commands:
 S - Show  R - Run (batch)   D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit  B - Checkpoint
 ? - Show all line commands

Work Statement List: WSL.LIST

Sel Name     Created    Changed          ID       Restart
    *        *          *                *        *
--- -------- ---------- ---------------- -------- --------
    C0000023 2013/04/05 2013/04/05 09:15 VNDDHG
    DG29608  2013/04/05 2013/04/05 14:17 VNDDHG
    D1026676 2013/03/27 2013/03/27 10:03 VNDDHG
    D25359   2013/03/27 2013/03/27 10:47 VNDDHG
    REO26676 2013/03/18 2013/03/18 14:04 VNDDHG
    S28654   2013/02/28 2013/03/28 12:38 VNDDHG   Y
```

Figure 298. Work Statement List Library panel (ADB2W1)

The JCL to generate the batch job to produce the Validate Work Statement List report is displayed.

5. Submit the JCL.

   The Validate Work Statement List report is generated and displayed, as shown in the following figure.



```
 -------------------------------------------------------------------------------
 SDSF OUTPUT DISPLAY NBRONV   J0086325  DSID   105 LINE 1       COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> PAGE
--------------------------------------------------------------------------------
ADB2WVL - Validate Work Statement List
--------------------------------------------------------------------------------

          DB2 Administration Tool
          5697-L90 (C) Copyright IBM Corporation 2001, 2005.
          All rights reserved.  Licensed materials - property of IBM.
          US Government Users Restricted Rights - Use, duplication or disclosure
          restricted by GSA ADP schedule contract with IBM Corp.

--------------------------------------------------------------------------------
REFERENCE FOR CATALOG OBJECT STATUS
--------------------------------------------------------------------------------
IMPLICITLY DROPPED OBJECTS - Existing catalog objects that are implicitly
                             dropped and not recreated by the WSL.
TEMPORARY OBJECTS          - Objects that are created and dropped during
                             execution of the WSL.  Temporary objects do not
                             exist in the catalog before or after WSL execution.
CREATED OBJECTS            - Objects that are created by the WSL that did not
                             exist in the catalog.
EXPLICITLY DROPPED OBJECTS - Existing catalog objects that are explicitly
                             dropped and not recreated by the WSL.
ALTERED OBJECTS            - Existing catalog objects that are modified by
                             ALTER statements in the WSL.
RECREATED OBJECTS          - Existing catalog objects that are implicitly or
                             explicilty dropped and later recreated by the WSL.
--------------------------------------------------------------------------------


VALIDATE WORK STATEMENT LIST REPORT
===================================

Prepared on DSN7 (DB2 Release 720) by NBRON at 2006-07-08 10:48
for NBRON.WLIST.VALIDATE(SAMPLE)

SQL error in PREPARE for statement:
 CREATE SEQUENCE ORDER_SEQ             START WITH 1             INCREMEN
 DSNT408I SQLCODE = -104, ERROR:  ILLEGAL SYMBOL "START". SOME SYMBOLS THAT
          MIGHT BE LEGAL ARE: FOR
 DSNT418I SQLSTATE   = 42601 SQLSTATE RETURN CODE
 DSNT415I SQLERRP    = DSNHPARS SQL PROCEDURE DETECTING ERROR
 DSNT416I SQLERRD    = 0  0  0  -1  40  0 SQL DIAGNOSTIC INFORMATION
 DSNT416I SQLERRD    = X'00000000'  X'00000000'  X'00000000'  X'FFFFFFFF'
          X'00000028'  X'00000000' SQL DIAGNOSTIC INFORMATION

Error processing Database  ABCDE in a ALTER statement:Object does not exist
Error processing Table DSN8720.ABCDTB in a ALTER statement:Object does not exist
Error processing Table DSN8720.DEPT in a ALTER statement:Object does not exist
Error processing Table DSN8720.ABCDTB in a ALTER statement:Object does not exist
Error processing Index DSN8720.ABCDIX in a ALTER statement:Object does not exist
Error processing Index DSN8720.XDEPT1 in a ALTER statement:Object does not exist
Error processing Sequence NBRON.org_seq in a ALTER statement:Object does not exist
Error processing Sequence VNDSHL2.SEQ14 in a CREATE statement:Object already exists
Error processing Sequence VNDSHL2.SEQ13 in a DROP statement:Object does not exist
.
.
.
```

Figure 299. Validate Work Statement List report (1 of 2)

```
.
.
.
IMPLICITLY DROPPED OBJECTS
--------------------------
  Referential constraint AHXTOOLS.PROJACT
  Referential constraint AHXTOOLS.PROJ
  Referential constraint AHXTOOLS.PROJ
  Referential constraint AHXTOOLS.DEPT
  Referential constraint AHXTOOLS.PROJ
  Referential constraint AHXTOOLS.PROJACT
  Referential constraint AHXTOOLS.DEPT

ALTERED OBJECTS
---------------
  Function NBRON.SPECIFICFFFF1


TEMPORARY OBJECTS
-----------------
  Sequence NBRON.org_seq
  Table Space DSN8D72A.DSN8S72D
  Table DSN8720.DEPT
  Table DSN8720.ABCDTB


CREATED OBJECTS
----------------
  Table NBRON.TBDSN80


RECREATED OBJECTS
------------------
  Table QUADPB02.TBADPB02
  Table Space DBADPB02.TPADPB01
  View QUADPB02.VWADPB02
  View QUADPB02.VWADPB04
  View QUADPB02.VWADPB05
  View QUADPB02.VWADPB06
  View QUADPB02.VWADPB09
  View QUADPB02.VWADPB12
  View QUADPB02.VWADPB14
  View QUADPB02.VWADPB15
  View QUADPB02.VWADPB16
  View QUADPB02.VWADPB17
  View QUADPB02.VWADPB18
  View QUADPB02.VWADPB19
  Index QUADPB02.IPADPB01
  Index QUADPB02.IPADPB02
  Referential constraint QUADPB02.TBADPB02 QUADPB02.TBADPB01 FKADPB03
  Referential constraint QUADPB02.TBADPB01 QUADPB02.TBADPB02 FKADPB02
  Referential constraint QUADPB02.TBADPB04 QUADPB02.TBADPB02 FKADPB04
  Referential constraint QUADPB02.TBADPB05 QUADPB02.TBADPB02 FKADPB07
```

Figure 300. Validate Work Statement List report (2 of 2)

What to do next

After the initial validation completes, you can impose additional custom validation rules to Db2 objects.

- For Object Compare and CM Analyze, use the STMTEXIT parameter. The value for this parameter can be found on the Options for Change Functions panel (ADB2PCO). You can manually update that value by changing it in the JCL, then submitting the JCL.

```
//ADBWL    DD DISP=SHR,
//         DSN=SYSADM.WSL
//CAT      DD DSN=&&CATOUT2,
//         DCB=(LRECL=16800,RECFM=VB,DSORG=PS),
//         SPACE=(TRK,(15,15),RLSE),
//         UNIT=SYSDA,
//         DISP=(NEW,PASS)
//ADBUEXEC DD DISP=SHR,DSN=SPF.PRODUCT.ISPTLIB
//ADBUEXE1 DD DISP=SHR,DSN=SPF.PRODUCT.ISPTLIB
//VALOUT   DD SYSOUT=*
//IN       DD *
  DB2SYS    = DSNB,
  DB2ALOC   =
  DB2SERV   = DSNB
  DB2AUTH   = 'S22957'
  DB2REL    = 1115
  ADBTEST   = YES,
  PLAN      = ADB,
  ADBASUSR  = ,
  ADBASUSB  = NO,
  SRCWSLST  = TEST3,
  SRCWSLIB  = SYSADM.WSL,
  STMTEXIT  = TEST
/*
/*
```

- For CM Batch, use the VALIDATION_STMTEXIT parameter. This parameter does not have a default value.

```
//* DB2 ADMIN ISPF BATCH
//*
//T03REG EXEC GOCCM,SSID=DSNB,PLAN=ADBDEV,GRP=UB2DEV0,USRGRP=S45801
//GOCCM.PARMS  DD *
 ACTION_IMPORT_CHANGE='N'
 ACTION_ANALYZE_CHANGE='Y'
 CHANGE_OWNER='S22957'
 CHANGE_NAME='IMPORT_DSTCHANGE'
 PDS_FOR_WSL='SYSADM.WSL'
 VALIDATE_STMTEXIT='TEST'
 CHANGE_COMMENT=''
 ;
```

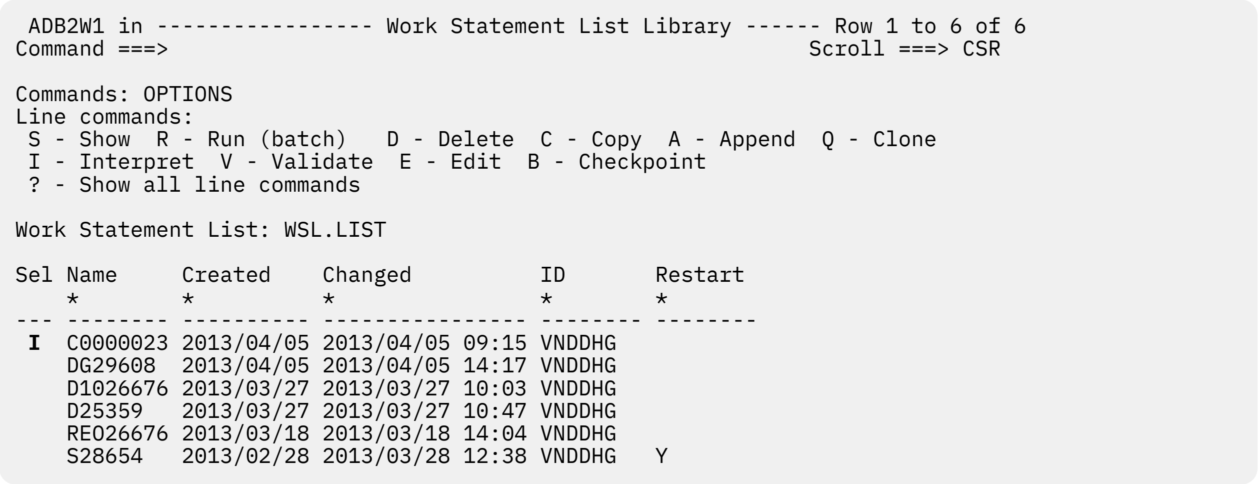
Interpreting a WSL

Before you run a work statement list (WSL), you might want to check the contents to see the types of statements that it contains. Interpreting a WSL allows you to generate a report that selectively lists the different SQL statements, Db2 commands, and utility statements in the WSL.

Procedure

To interpret a WSL:

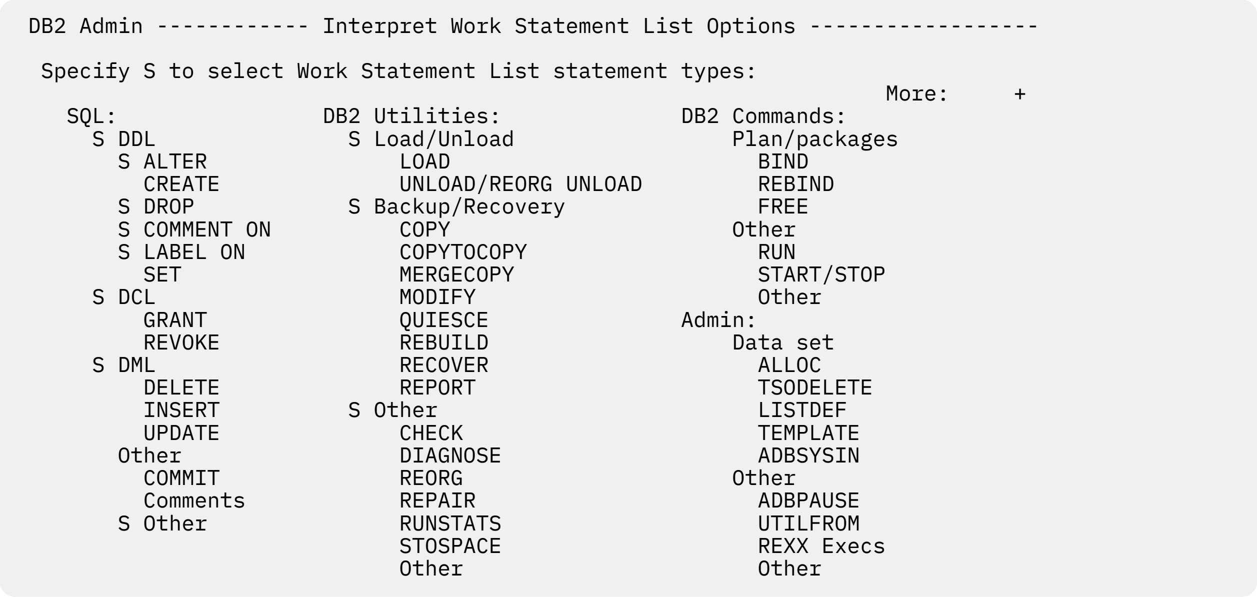
1. On the **DB2 Administration Menu (ADB2)** panel, specify option W, and press Enter.
2. On the **Manage Work Statement Lists (ADB2W)** panel, in the **Work stmt list dsn** field, specify the data set name of the ISPF library that contains the WSL.
3. Specify option 1, and press Enter.
4. On the **Work Statement List Library (ADB2W1)** panel, issue the I line command against the WSL, and press Enter.



```
 ADB2W1 in ----------------- Work Statement List Library ------ Row 1 to 6 of 6
Command ===>                                                  Scroll ===> CSR

Commands: OPTIONS
Line commands:
 S - Show  R - Run (batch)   D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit  B - Checkpoint
 ? - Show all line commands

Work Statement List: WSL.LIST

Sel Name     Created    Changed          ID       Restart
    *        *          *                *        *
--- -------- ---------- ---------------- -------- --------
 I  C0000023 2013/04/05 2013/04/05 09:15 VNDDHG
    DG29608  2013/04/05 2013/04/05 14:17 VNDDHG
    D1026676 2013/03/27 2013/03/27 10:03 VNDDHG
    D25359   2013/03/27 2013/03/27 10:47 VNDDHG
    REO26676 2013/03/18 2013/03/18 14:04 VNDDHG
    S28654   2013/02/28 2013/03/28 12:38 VNDDHG   Y
```

Figure 301. Work Statement List Library panel (ADB2W1)

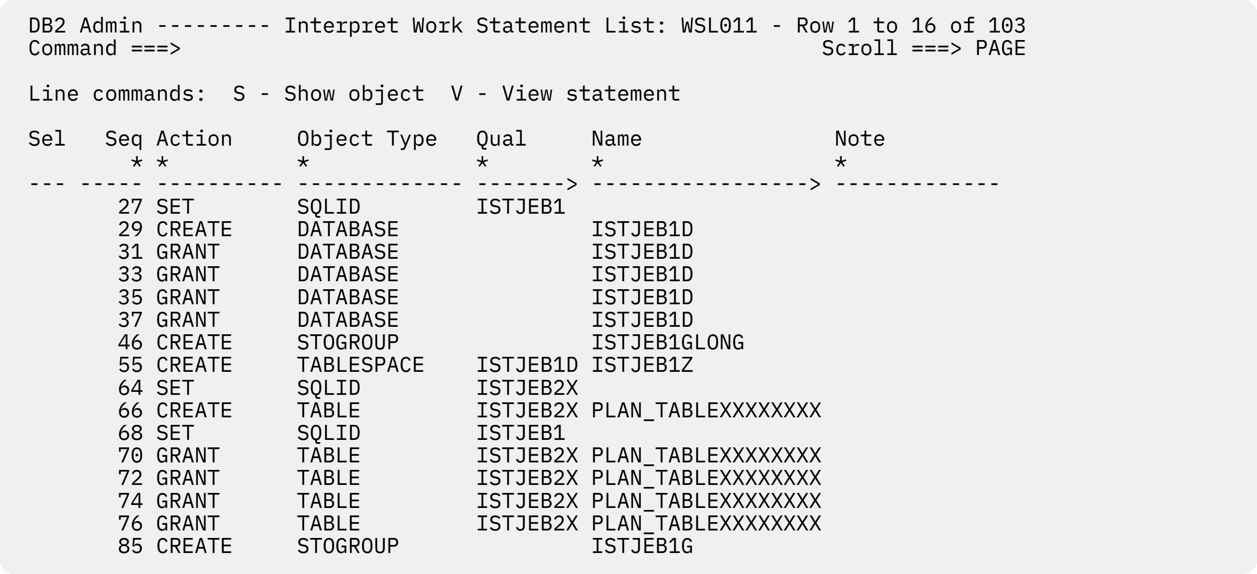
5. On the **Interpret Work Statement List Options (ADB2W1O)** panel, specify an S next to the statement types that you want interpreted, and press Enter:



```
DB2 Admin ------------ Interpret Work Statement List Options -----------------

 Specify S to select Work Statement List statement types:
                                                                 More:     +
   SQL:               DB2 Utilities:            DB2 Commands:
     S DDL              S Load/Unload               Plan/packages
       S ALTER              LOAD                      BIND
         CREATE             UNLOAD/REORG UNLOAD       REBIND
       S DROP           S Backup/Recovery             FREE
       S COMMENT ON         COPY                    Other
       S LABEL ON           COPYTOCOPY                RUN
         SET                MERGECOPY                 START/STOP
     S DCL                  MODIFY                    Other
         GRANT              QUIESCE             Admin:
         REVOKE             REBUILD                 Data set
     S DML                  RECOVER                   ALLOC
         DELETE             REPORT                    TSODELETE
         INSERT         S Other                       LISTDEF
         UPDATE             CHECK                     TEMPLATE
       Other                DIAGNOSE                  ADBSYSIN
         COMMIT             REORG                   Other
         Comments           REPAIR                    ADBPAUSE
       S Other              RUNSTATS                  UTILFROM
                            STOSPACE                  REXX Execs
                            Other                     Other
```

*Figure 302. **Interpret Work Statement List Options (ADB2W1O)** panel*

The **Interpret Work Statement List (ADB2W1I)** panel displays information about the selected statement types:



```
DB2 Admin --------- Interpret Work Statement List: WSL011 - Row 1 to 16 of 103
Command ===>                                                Scroll ===> PAGE

Line commands:  S - Show object  V - View statement

Sel   Seq Action     Object Type   Qual     Name               Note
        * *          *             *        *                  *
--- ----- ---------- ------------ -------> ----------------> ------------
       27 SET        SQLID         ISTJEB1
       29 CREATE     DATABASE               ISTJEB1D
       31 GRANT      DATABASE               ISTJEB1D
       33 GRANT      DATABASE               ISTJEB1D
       35 GRANT      DATABASE               ISTJEB1D
       37 GRANT      DATABASE               ISTJEB1D
       46 CREATE     STOGROUP               ISTJEB1GLONG
       55 CREATE     TABLESPACE    ISTJEB1D ISTJEB1Z
       64 SET        SQLID         ISTJEB2X
       66 CREATE     TABLE         ISTJEB2X PLAN_TABLEXXXXXXXX
       68 SET        SQLID         ISTJEB1
       70 GRANT      TABLE         ISTJEB2X PLAN_TABLEXXXXXXXX
       72 GRANT      TABLE         ISTJEB2X PLAN_TABLEXXXXXXXX
       74 GRANT      TABLE         ISTJEB2X PLAN_TABLEXXXXXXXX
       76 GRANT      TABLE         ISTJEB2X PLAN_TABLEXXXXXXXX
       85 CREATE     STOGROUP               ISTJEB1G
```

Figure 303. Interpret Work Statement List report

Use the S line command to view more information about an object. This line command is valid for only those objects that are in the Db2 catalog, such as databases, table spaces, and indexes.

Running a WSL

You can run a work statement list (WSL) either in batch or online.

About this task

When you run a WSL, it invokes the batch restart program, ADBTEP2.

Recommendation: For a WSL that was generated to implement changes through Change Management, do not run the WSL from the **Work Statement List Library (ADB2W1)** panel. Instead, use the RN command on the **CM - Changes (ADB2C11)** panel to run the change, which causes the WSL to be run. Using the

RN command is recommended, because any change that you want to track through Change Management must be made through Change Management. Also, do not use the line commands on the **Work Statement List Library (ADB2W1)** panel to edit, delete, copy, append, or clone the WSL.

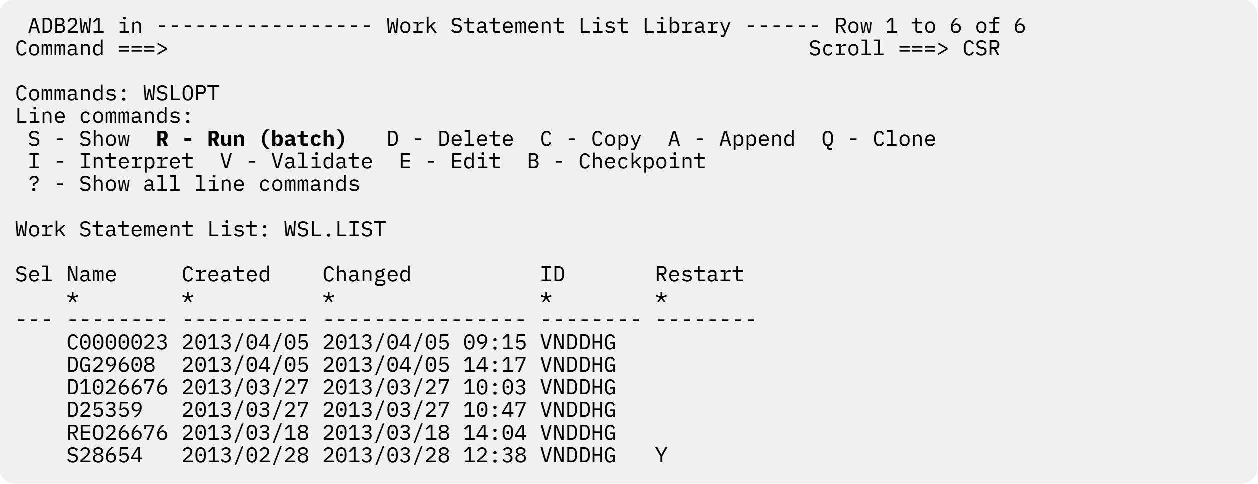
Procedure

Note: If you have already built the JCL to run the WSL, skip to step "6.b" on page 554. For example, you might have built this JCL when running Object Comparison Tool.

To run a WSL:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option W, and press Enter.
2. On the **Manage Work Statement Lists (ADB2W)** panel, in the **Work stmt list dsn** field, specify the data set name of the ISPF library that contains the WSL.
3. Specify option 1, and press Enter.

   The **Work Statement List Library (ADB2W1)** panel lists the work statement lists in the specified library:



```
 ADB2W1 in ----------------- Work Statement List Library ------ Row 1 to 6 of 6
Command ===>                                                  Scroll ===> CSR

Commands: WSLOPT
Line commands:
 S - Show  R - Run (batch)   D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit  B - Checkpoint
 ? - Show all line commands

Work Statement List: WSL.LIST

Sel Name     Created    Changed          ID       Restart
    *        *          *                *        *
--- -------- ---------- ---------------- -------- --------
    C0000023 2013/04/05 2013/04/05 09:15 VNDDHG
    DG29608  2013/04/05 2013/04/05 14:17 VNDDHG
    D1026676 2013/03/27 2013/03/27 10:03 VNDDHG
    D25359   2013/03/27 2013/03/27 10:47 VNDDHG
    RE026676 2013/03/18 2013/03/18 14:04 VNDDHG
    S28654   2013/02/28 2013/03/28 12:38 VNDDHG   Y
```

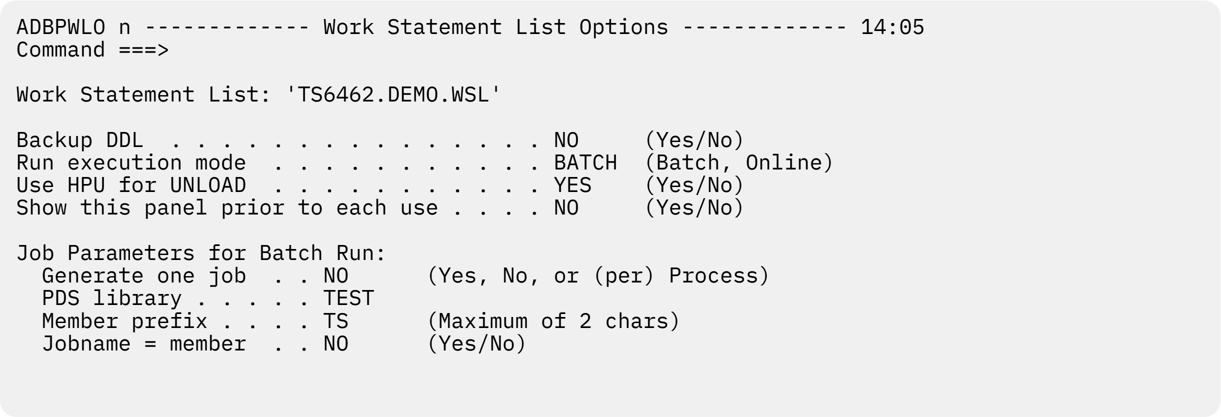
   Figure 304. ***Work Statement List Library (ADB2W1)*** *panel*

   Notice that the R line command lists a run method. For example, in the preceding screen the R line command will run the WSL in batch.

   In the following example, the R line command will run the WSL online:

```
Line commands:
 S - Show  R - Run (online)  D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit  B - Checkpoint
```

4. If you want to change any WSL options, including the run method that is listed next to the R line command, take the following actions:
   a) Issue the WSLOPT command, and press Enter.
   b) On the **Work Statement List Options (ADBPWLO)** panel, change any options and job parameters that you want, and press Enter.



```
ADBPWLO n ------------ Work Statement List Options ------------ 14:05
Command ===>

Work Statement List: 'TS6462.DEMO.WSL'

Backup DDL  . . . . . . . . . . . . . . . NO     (Yes/No)
Run execution mode  . . . . . . . . . . . BATCH  (Batch, Online)
Use HPU for UNLOAD  . . . . . . . . . . . YES    (Yes/No)
Show this panel prior to each use . . . . NO     (Yes/No)

Job Parameters for Batch Run:
  Generate one job  . . NO      (Yes, No, or (per) Process)
  PDS library . . . . . TEST
  Member prefix . . . . TS      (Maximum of 2 chars)
  Jobname = member  . . NO      (Yes/No)
```

*Figure 305. **Work Statement List Options (ADBPWLO)** panel*

For example, you can set the run method to batch or online in the **Run execution mode** field, and you can specify whether to use Db2 High Performance Unload (HPU) as the unload method.

If you specify **Generate one job** = YES, a single batch job is generated from the WSL that combines all the processes. You do not need to specify values for the remaining fields on the panel. If you specify **Generate one job** = NO, enter the library name in the **PDS library** field, the member prefix name in the **Member prefix** field, and whether the job name is the same as the member name in the **Jobname = member** field. Multiple jobs are generated and saved in the specified job library PDS.

Restriction: The maximum allowable prefix length is seven characters. When naming the member prefix, the number of characters that is required by the system to represent multiple jobs determines how many characters you can use to name the prefix. The max chars field is populated dynamically with the number of characters that are available after the number of jobs is established. For example, running 57 jobs uses two of the seven allowable characters (01, 02...57), which leaves five characters for you to use for the prefix name. Therefore, the number five is populated in the max chars field. When you specify job parameters while you are building a batch job to run the work statement list, the member prefix length is limited to a maximum of four characters.

5. On the **Work Statement List Library (ADB2W1)** panel, specify the R line command next to the WSL that you want to run.

   If you chose to run the WSL online, ADBTEP2 is run online and all input instructions are processed sequentially. When you run a WSL online, certain program or utilities that are intended to be run in batch might issue messages to the terminal. Make note of these messages, and press Enter to clear the messages.

   If you chose to run the WSL in batch, the JCL to generate the batch job is displayed.

6. If you are running the WSL in batch:

   a) Submit the JCL to generate the batch jobs.

      One or more jobs are created. Each job includes a step to run the ADBTEP2 and the job's set of input instructions (batch statement list) for ADBTEP2.

      If the WSL includes the PARALLEL command, Db2 Admin Tool generates multiple jobs with the following job names:

      ```
      <prefix><m><seqnumber>
      ```

      <prefix>
      A specified prefix. The prefix can be from 4 to 6 characters, depending on the number of parallel jobs.

<m>

The first character in the word following the PARALLEL command. For example, U for UNLOAD; R for RELOAD.

<seqnumber>

The generated sequence number. The sequence number can be from 1 to 3 characters (*n* to *nnn*), depending on the number of parallel jobs:

n

For 1 to 9 parallel jobs

nn

For 10 to 99 parallel jobs

nnn

For more than 99 parallel jobs

The maximum length of a job name is 8 characters.

b) Run the batch jobs.

You must submit these jobs in the sequence that they are presented in the following table.

Table 30. Work statement list submission sequence for multiple operations. The following table shows examples of different jobs and what action is taken.

Job details	Job action
*prefix*U001 *prefix*U002 .... *prefix*U*nnn*	One job is created for each UNLOAD operation.
*prefix*2	One job is created for all DROP, CREATE, and ALTER operations.
*prefix*L001 *prefix*L002 .... *prefix*L*nnn*	One job is created for each RELOAD operation.
*prefix*3	One job is created for all utilities except REORG.
*prefix*R001 *prefix*R002 .... *prefix*R*nnn*	One job is created for each REORG operation.

If the jobs fail, you can restart them. See "Restarting a WSL" on page 556

What to do next

If the WSL included a LOAD operation, review the Load Summary Report in LOADRPT. This report indicates whether records were discarded when data was loaded. When a Load Summary Report step exists, SYSPRINT output from the preceding ADBTEP2 step is recorded in ADBPRINT of the Load Summary Report step. If the WSL does not include a LOAD operation, ADBTEP2 messages are recorded in SYSPRINT of the ADBTEP2 step.

Load summary report

Checking the load summary report (located in LOADPRT) at the end of a WSL run is easier than scanning the WSL execution log and checking for instances of load-generated discard records.

The load summary report helps you ensure that no data was unexpectedly lost.

The load summary report contains the following information:

- The name of the object
- The number of input records
- The number of records that were loaded
- The number of records that were discarded

The example in the following figure shows a load summary report in which the number of input and loaded records for three tables were the same, but records were discarded for another table.

```
15697-L90 IBM DB2 Administration Tool for Z/OS       Load Summary Report for Worklist(ST8)


Table owner        Table name               Input      Loaded   Discarded Status
------------------ ------------------ ----------- ----------- ----------- --------
"SYSADM"           "TBADAS01"                1255        1255           0 ********
"SYSADM"           "TBADAS02"                 855         799          56 discards
"SYSADM"           "TBADAS03"                2033        2033           0 ********
"SYSADM"           "TBADAS04"                1444        1444           0 ********
```

Figure 306. Example of load summary report

When the report contains a large number of rows, you will need to scroll through the report to see all of the information in the report. When the table name exceeds the number of characters that can be displayed in the **Table Name** field, a footnote suffix is added to the table name, and the full table name is displayed at the bottom of the report. The following example shows the format that is used to display long table names:

```
15697-L90 IBM DB2 Administration Tool for Z/OS       Load Summary Report for Worklist(ST9)


Table owner        Table name               Input      Loaded   Discarded Status
------------------ ------------------ ----------- ----------- ----------- --------
"SYSADM"           "TBADAS0190123(*1)        1006        1006           0 ********
"SYSADM"           "TBADAS0290123(*2)          75          75           0 ********
"SYSADM"           "TBADAS0390123(*3)        4031        4031           0 ********
"SYSADM"           "TBADAS0490123(*4)        2444        2444           0 ********

Footnotes:

(*1)

 "TBADAS01901234567890123456789O"

(*2)

 "TBADAS02901234567890123456789O"

(*3)

 "TBADAS03901234567890123456789O"

(*4)

 "TBADAS04901234567890123456789O"
```

Figure 307. Example of load summary report with long table names

Restarting a WSL

If a work statement list (WSL) fails in the middle of execution, you can restart it at the point that it failed or at a point that you specify. A WSL uses the restart capability of ADBTEP2.

Before you begin

- Correct any errors in the WSL or any other conditions that caused the failure.
- Decide which type of restart to use:

  System-controlled
  A system-controlled restart is automated. Db2 Admin Tool restarts the WSL from the point where it failed.

  User-controlled
  A user-controlled restart allows you to restart the WSL from a point different than where it failed.

- If you want to do a user-controlled restart, you must add a restart point to your WSL. To add a restart statement:

  1. View the WSL.
  2. On the **Show Work Statement List (ADB2W1S)** panel, specify the I line command next to the statement after which you want to insert a restart point, and press Enter.
  3. On the **Insert Work Statement (ADB2W1SI)** panel, specify option 9 and specify a name for the restart point next to option 9.

```
ADB2W1SI  ------------------ DC1A Insert Work Statement ----------------- 16:59
Option ===> 9

Enter the type of statement you want to insert:

 1 - A comment statement
 2 - A definition SQL statement
 3 - A authorization SQL statement
 4 - A update SQL statement
 5 - A DB2 command
 6 - A DSN command
 7 - A DB2 utility statement
 8 - A DB2 Admin statement
 9 - A restart statement          RSTPOINT1
```

For more information about the RESTART statement and the allowable values for the restart point name, see "RESTART" on page 534.

Alternatively you can specify the restart point directly in the batch statement list. Add the line `--#RESTART <string>` at the point that you want the statement to restart from. In addition, you must add `RESTART AT <string>` with a terminator as the first line, like the following example:

```
//SYSIN DD *
RESTART AT BOB;
DROP TABLE
    POSTO.PPP1;
COMMIT;
--#RESTART BOB
DROP TABLE POSTO.PPP2;
COMMIT;
```

You can add as many user-defined restart points to a WSL as you want, but only one will be used for restart.

About this task

The Batch Restart program (ADBTEP2) enables you to restart or resume the execution of an input stream at an intermediate point if any one of the statements in that stream should fail. The process involves creating or updating a record in a checkpoint table each time that a COMMIT statement is encountered in the input stream. By using this table, execution can be resumed with the first statement following the last successful commit point before the failure, bypassing all prior successfully executed statements.

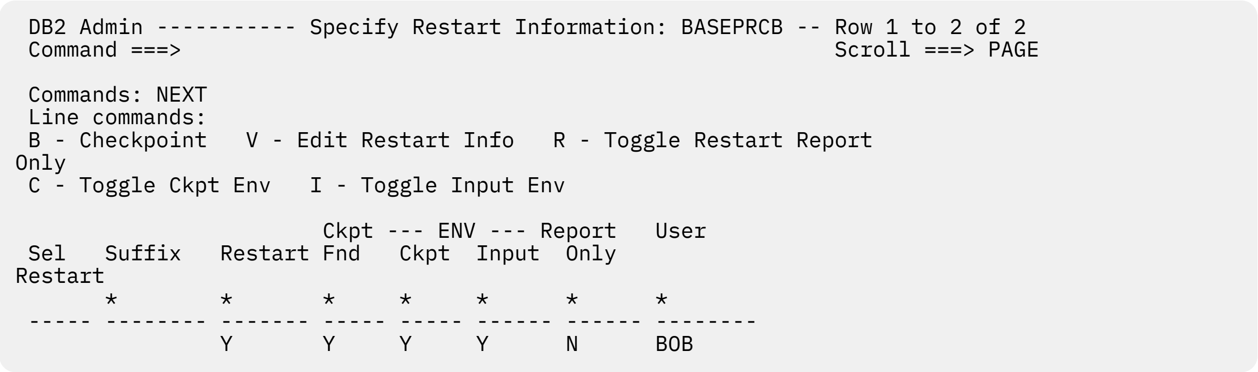
Procedure

To restart a failed WSL:

1. On the **Work Statement List Library (ADB2W1)** panel, reissue the R line command against the WSL.

   Requirement: If the WSL contains parallel processing (the WSL includes PARALLEL statements), the WSL must be restarted in the same mode that it was originally run (either online or batch).

   The **Specify Restart Information** panel is displayed:



```
 DB2 Admin ----------- Specify Restart Information: BASEPRCB -- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: NEXT
 Line commands:
 B - Checkpoint   V - Edit Restart Info   R - Toggle Restart Report
Only
 C - Toggle Ckpt Env   I - Toggle Input Env

                        Ckpt --- ENV --- Report   User
 Sel   Suffix   Restart Fnd   Ckpt  Input  Only
Restart
       *        *       *     *     *      *      *
 ----- -------- ------- ----- ----- ------ ------ --------
                Y       Y     Y     Y      N      BOB
```

*Figure 308. **Specify Restart Information** panel*

2. Depending on the type of restart that you want, take the following actions on the **Specify Restart Information** panel:

Option	Description
System-controlled restart (default)	To restart the WSL from the point of failure: a. In the **Restart** column, specify Y. b. Issue the CONTINUE command.
User-controlled restart	To restart the WSL from a point that you specify: a. Issue the V line command. b. In the **Restart** column, enter U. c. In the **User Restart** column, specify the name of the restart point in your WSL or batch statement list. (This value is the value of *name* in `--#RESTART` *name*). d. Issue the CONTINUE command.
Restart report only	To simulate a restart without actually running a restart, so that you can see the results before deciding whether to run a restart: a. Issue the R line command next to the WSL that you want to restart. b. Issue the CONTINUE command.

Restarting a WSL that was run by another user

You can restart a WSL that was run by another user but did not complete successfully.

Procedure

1. Determine the user ID of the user who ran the WSL.

   You can find the user ID in the checkpoint table.

2. Issue the R (Run in batch) command on **Work Statement List Library** panel for the WSL that you want to restart.

   The JCL to generate the batch job is displayed.

3. Edit the batch job at the ADBTEP2 restart job step and specify the USER parameter with the user ID of the user who originally ran the WSL.
   For example, if a user with user ID SYSADM ran the WSL, the following portion of code shows how the edited JCL would look with the USER parameter added:

```
000036  RUN  PROGRAM(ADBTEP2) PLAN(ADBTEP2) -
000037    LIB('ADB.QA260.ISPLLIB')  -
000038    PARMS('/WORKLIST(JTKZ) SSID(V81A)  -
 ''''''    USER(SYSADM) -
000039  RESTART(YES),BINDERROR(MAXE)')
```

Important: The USER and CHANGEID parameters are mutually exclusive. Ensure that the JCL does not include a CHANGEID parameter.

4. Submit the JCL.

Converting the WSL format in CM batch

If you use Change Management (CM) batch, you can convert a work statement list (WSL) between internal ISPF table format and readable format. This conversion allows you to view and potentially edit a WSL without using Db2 Admin Tool panels.

About this task

When Db2 Admin Tool generates a WSL, it is stored in an internal format in ISPF tables. To view a WSL, you can use the WSL panels. However, if you are using CM batch instead of panels, you might want to convert a WSL to a readable format in your CM batch job. After you view and optionally edit the WSL, you can convert it back to the internal ISPF table format and run it.

Procedure

To convert the WSL format in CM batch:

- Specify one of the following CM batch parameters:

  action_convert_to_read_wsl
  Converts the WSL to readable format. For details, see "ACTION_CONVERT_TO_READ_WSL" on page 667.

  action_convert_to_ispf_wsl
  Converts the WSL to internal ISPF table format. For details, see "ACTION_CONVERT_TO_ISPF_WSL" on page 667.

  You can specify these action parameters regardless of the Db2 attachment facility you use (CAF or RRSAF).

- Specify the following CM batch parameters to identify the PDS member that has or will have the WSL in readable format:

  pds_for_wsl_conv
  PDS name. For details, see "PDS_FOR_WSL_CONV " on page 709.

  worklist_name_conv
  Member name. For details, see "WORKLIST_NAME_CONV" on page 744.

Example

The following example CM batch job imports and analyzes the input DDL and converts the WSL to readable format:

```
//GOCCM.PARMS DD *
 ,ACTION_IMPORT_CHANGE         = 'Y'
 ,ACTION_ANALYZE_CHANGE        = 'Y'
 ,ACTION_RUN_CHANGE            = 'N'
 ,ACTION_COMPARE               = 'N'
 ,ACTION_CONVERT_TO_READ_WSL   = 'Y'
 ,CHANGE_OWNER                 = 'userid'
 ,CHANGE_NAME                  = '&USERID. CHANGE &CURTS.'
 ,CHANGE_COMMENT               = ''
 ,CONTENT_OF_APPLY_JOBS        = 'D'
 ,EXISTING_DATA_SET_ACTION     = 'REPLACE'
 ,IMPORT_PENDING_CHANGE_ACTION = 'S'
 ,PREFIX_FOR_DATA_SETS         = 'userid.CM'
 ,RUN_SQLID                    = '<NONE>'
 ,VALIDATE_DDL                 = 'Y'
 ,WORKLIST_NAME_CONV           = 'READ001'
 ,PDS_FOR_WSL_CONV             = 'WSLCONV'
```

```
/*
//GOCCM.IMCHG001 DD *
CREATE TABLE schema.CNVTEST1
( COL1 INTEGER NOT NULL WITH DEFAULT )
IN DATABASE database;
```

After you view and potentially modify the WSL in *userid*.CM.WSLCONV(READ001), you can convert the WSL back to internal ISPF table format to run it, as shown in the following example:

```
//GOCCM.PARMS DD *
 ,ACTION_CONVERT_TO_ISPF_WSL   = 'Y'
 ,ACTION_RUN_CHANGE            = 'Y'
 ,CHANGE_OWNER                 = 'userid'
 ,CHANGE_NAME  = 'userid CHANGE 2019-01-01-10.41.16.817045'
 ,DO_RUNTIME_ANALYZE = 'N'
 ,WORKLIST_NAME_CONV = 'READ001'
 ,PDS_FOR_WSL_CONV = 'WSLCONV'
```

Note: The convert action (either converting to readable format or converting to ISPF table format) can be done stand-alone. It does not need to be specified with any other action parameter, such as action_analyze_change or action_run_change.

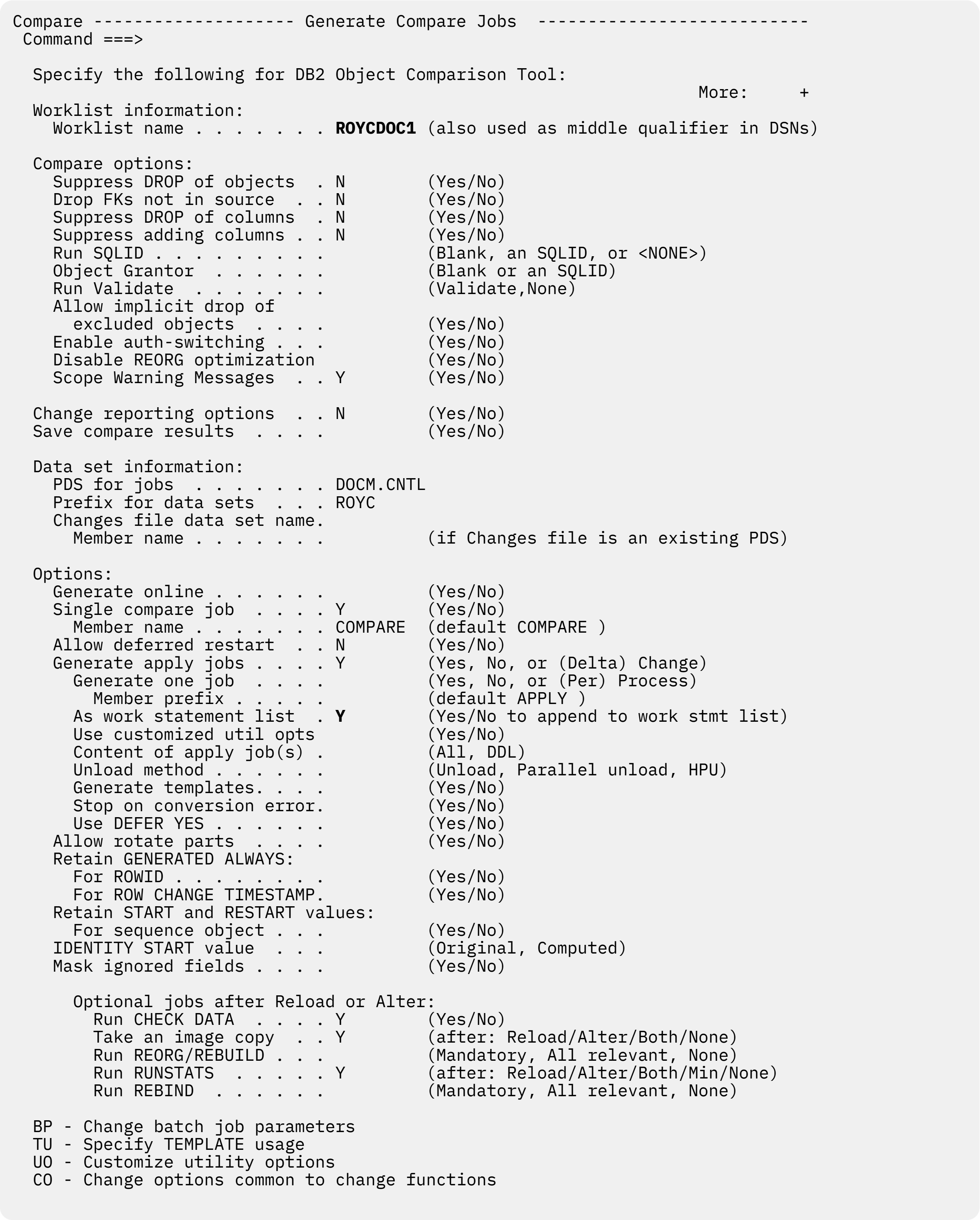
Sample scenario for creating and using a work statement list

This scenario shows how to use IBM Db2 Object Comparison Tool for z/OS to create a work statement list (WSL) and then run it.

Suppose that you have two databases, each with two tables. One of the databases is older and outdated, and you want to upgrade it to the new database. You can use Object Comparison Tool to generate the tasks that are required to upgrade the database and save those tasks in a WSL.

To create this WSL:

1. Define the required input and masks. (Options **1** through **4** on the **DB2 Object Comparison Tool Menu (GOCMENU)** panel)
2. From the **DB2 Object Comparison Tool Menu (GOCMENU)** panel, select option **5**.
3. On the **Generate Compare Jobs (GOC5)** panel:
   - Specify the new name of the WSL.
   - In the **As work statement list** field, specify Y to indicate that the job should be saved as a WSL.
   - Specify any other relevant options.



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

  Specify the following for DB2 Object Comparison Tool:
                                                                    More:     +
  Worklist information:
    Worklist name . . . . . . . ROYCDOC1 (also used as middle qualifier in DSNs)

  Compare options:
    Suppress DROP of objects  . N        (Yes/No)
    Drop FKs not in source  . . N        (Yes/No)
    Suppress DROP of columns  . N        (Yes/No)
    Suppress adding columns . . N        (Yes/No)
    Run SQLID . . . . . . . . .          (Blank, an SQLID, or <NONE>)
    Object Grantor  . . . . . .          (Blank or an SQLID)
    Run Validate  . . . . . . .          (Validate,None)
    Allow implicit drop of
      excluded objects  . . . .          (Yes/No)
    Enable auth-switching . . .          (Yes/No)
    Disable REORG optimization           (Yes/No)
    Scope Warning Messages  . . Y        (Yes/No)

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

  Data set information:
    PDS for jobs  . . . . . . . DOCM.CNTL
    Prefix for data sets  . . . ROYC
    Changes file data set name.
      Member name . . . . . . .          (if Changes file is an existing PDS)

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

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

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

Figure 309. ***Generate Compare Jobs (GOC5)*** *panel*

4. In the **Specify Work Statement List (ADB2WLDA)** panel, enter the data set in which to store the new WSL. If the data set does not exist, it is created.

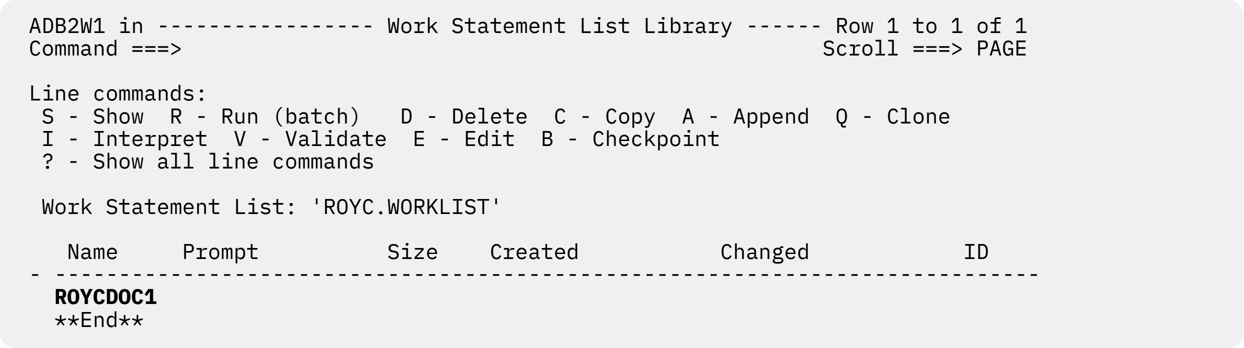
   A Object Comparison Tool JCL job is generated for this new WSL.

5. Submit this job to generate the WSL that can be used to upgrade the old tables to the new tables. Check the output to ensure that the job executed successfully.

6. From the **DB2 Administration Menu (ADB2)** panel, select option **W**.

7. On the **Manage Work Statement Lists (ADB2W)** panel, select option **1** to display a list of WSLs.

   The following list includes the new ROYDCOC1 WSL that you created.

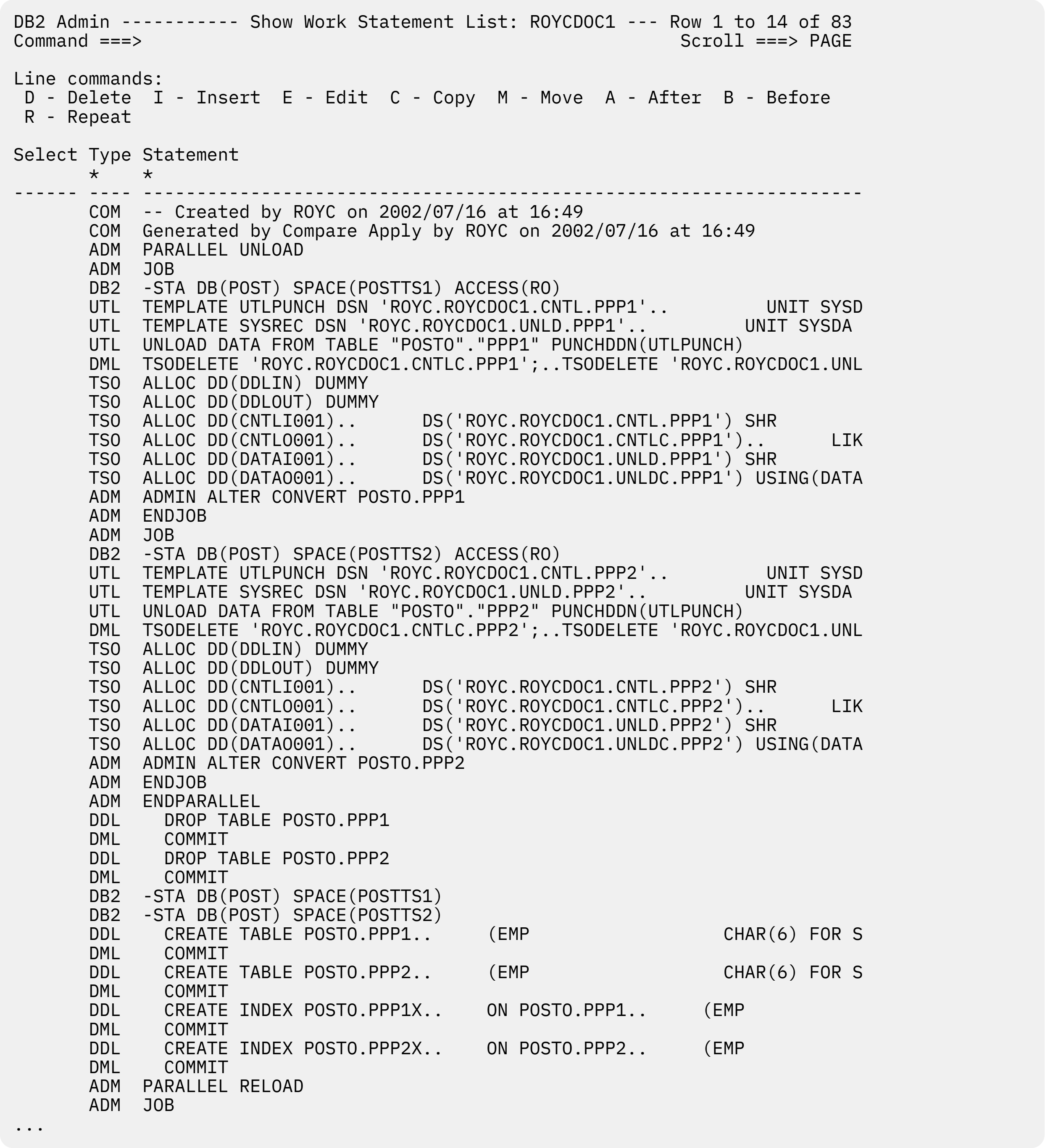


```
ADB2W1 in ----------------- Work Statement List Library ------ Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Line commands:
 S - Show  R - Run (batch)   D - Delete  C - Copy  A - Append  Q - Clone
 I - Interpret  V - Validate  E - Edit  B - Checkpoint
 ? - Show all line commands

 Work Statement List: 'ROYC.WORKLIST'

   Name     Prompt          Size    Created           Changed            ID
- ------------------------------------------------------------------------------
  ROYCDOC1
  **End**
```

*Figure 310. **Work Statement List Library (ADB2W1)** panel*

8. Use the S line command to show the contents of the new WSL.

   In the WSL, the TYPE column specifies the statement type (such as DDL statement, Db2 command, or Db2 utility) for statements that are placed in the batch statement list when running the WSL. A value of ADM in the TYPE column indicates a control statement that can control the number of jobs that are created when the WSL is run.



```
DB2 Admin ----------- Show Work Statement List: ROYCDOC1 --- Row 1 to 14 of 83
Command ===>                                                  Scroll ===> PAGE

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat

Select Type Statement
       *    *
------ ---- -------------------------------------------------------------------
       COM  -- Created by ROYC on 2002/07/16 at 16:49
       COM  Generated by Compare Apply by ROYC on 2002/07/16 at 16:49
       ADM  PARALLEL UNLOAD
       ADM  JOB
       DB2  -STA DB(POST) SPACE(POSTTS1) ACCESS(RO)
       UTL  TEMPLATE UTLPUNCH DSN 'ROYC.ROYCDOC1.CNTL.PPP1'..         UNIT SYSD
       UTL  TEMPLATE SYSREC DSN 'ROYC.ROYCDOC1.UNLD.PPP1'..         UNIT SYSDA
       UTL  UNLOAD DATA FROM TABLE "POSTO"."PPP1" PUNCHDDN(UTLPUNCH)
       DML  TSODELETE 'ROYC.ROYCDOC1.CNTLC.PPP1';..TSODELETE 'ROYC.ROYCDOC1.UNL
       TSO  ALLOC DD(DDLIN) DUMMY
       TSO  ALLOC DD(DDLOUT) DUMMY
       TSO  ALLOC DD(CNTLI001)..      DS('ROYC.ROYCDOC1.CNTL.PPP1') SHR
       TSO  ALLOC DD(CNTLO001)..      DS('ROYC.ROYCDOC1.CNTLC.PPP1')..      LIK
       TSO  ALLOC DD(DATAI001)..      DS('ROYC.ROYCDOC1.UNLD.PPP1') SHR
       TSO  ALLOC DD(DATAO001)..      DS('ROYC.ROYCDOC1.UNLDC.PPP1') USING(DATA
       ADM  ADMIN ALTER CONVERT POSTO.PPP1
       ADM  ENDJOB
       ADM  JOB
       DB2  -STA DB(POST) SPACE(POSTTS2) ACCESS(RO)
       UTL  TEMPLATE UTLPUNCH DSN 'ROYC.ROYCDOC1.CNTL.PPP2'..         UNIT SYSD
       UTL  TEMPLATE SYSREC DSN 'ROYC.ROYCDOC1.UNLD.PPP2'..         UNIT SYSDA
       UTL  UNLOAD DATA FROM TABLE "POSTO"."PPP2" PUNCHDDN(UTLPUNCH)
       DML  TSODELETE 'ROYC.ROYCDOC1.CNTLC.PPP2';..TSODELETE 'ROYC.ROYCDOC1.UNL
       TSO  ALLOC DD(DDLIN) DUMMY
       TSO  ALLOC DD(DDLOUT) DUMMY
       TSO  ALLOC DD(CNTLI001)..      DS('ROYC.ROYCDOC1.CNTL.PPP2') SHR
       TSO  ALLOC DD(CNTLO001)..      DS('ROYC.ROYCDOC1.CNTLC.PPP2')..      LIK
       TSO  ALLOC DD(DATAI001)..      DS('ROYC.ROYCDOC1.UNLD.PPP2') SHR
       TSO  ALLOC DD(DATAO001)..      DS('ROYC.ROYCDOC1.UNLDC.PPP2') USING(DATA
       ADM  ADMIN ALTER CONVERT POSTO.PPP2
       ADM  ENDJOB
       ADM  ENDPARALLEL
       DDL    DROP TABLE POSTO.PPP1
       DML    COMMIT
       DDL    DROP TABLE POSTO.PPP2
       DML    COMMIT
       DB2  -STA DB(POST) SPACE(POSTTS1)
       DB2  -STA DB(POST) SPACE(POSTTS2)
       DDL    CREATE TABLE POSTO.PPP1..     (EMP                  CHAR(6) FOR S
       DML    COMMIT
       DDL    CREATE TABLE POSTO.PPP2..     (EMP                  CHAR(6) FOR S
       DML    COMMIT
       DDL    CREATE INDEX POSTO.PPP1X..    ON POSTO.PPP1..     (EMP
       DML    COMMIT
       DDL    CREATE INDEX POSTO.PPP2X..    ON POSTO.PPP2..     (EMP
       DML    COMMIT
       ADM  PARALLEL RELOAD
       ADM  JOB
...
```

Figure 311. The contents of the new WSL

```
...
       UTL  TEMPLATE SYSREC DSN 'ROYC.ROYCDOC1.UNLDC.PPP1' DISP(SHR)
       UTL  TEMPLATE UTLDISC DSN 'ROYC.ROYCDOC1.SDISC.PPP1'..         UNIT SYSD
       UTL  TEMPLATE UTLERR DSN 'ROYC.ROYCDOC1.SERR.PPP1'..         UNIT SYSDA.
       UTL  TEMPLATE UTLMAP DSN 'ROYC.ROYCDOC1.SMAP.PPP1'..         UNIT SYSDA
       UTL  TEMPLATE UTLUT1 DSN 'ROYC.ROYCDOC1.SUT1.PPP1'..         UNIT SYSDA
       UTL  TEMPLATE UTLOUT DSN 'ROYC.ROYCDOC1.SOUT.PPP1'..         UNIT SYSDA
       UTL  UTILFROM ROYC.ROYCDOC1.CNTLC.PPP1..         ADD(SORTNUM 8 SORTDEVT
       UTL  TEMPLATE UTLUT1 DSN 'ROYC.ROYCDOC1.SUT1.PPP1'..         UNIT SYSDA
       UTL  TEMPLATE UTLOUT DSN 'ROYC.ROYCDOC1.SOUT.PPP1'..         UNIT SYSDA
       UTL  TEMPLATE UTLERR DSN 'ROYC.ROYCDOC1.SERR.PPP1'..         UNIT SYSDA
       UTL  CHECK DATA TABLESPACE POST.POSTTS1..  ERRDDN(UTLERR) WORKDDN(UTLUT1
       UTL  RUNSTATS TABLESPACE POST.POSTTS1..  TABLE("POSTO"."PPP1")..  INDEX(
       UTL  TEMPLATE SYSCOPY DSN 'ROYC.DSN7.IC.POST.POSTTS1(+1)'..         UNIT
       UTL  COPY TABLESPACE POST.POSTTS1 COPYDDN(SYSCOPY)
       UTL  MODIFY RECOVERY TABLESPACE POST.POSTTS1 DSNUM ALL..  DELETE AGE(35)
       ADM  ENDJOB
       ADM  JOB
       UTL  TEMPLATE SYSREC DSN 'ROYC.ROYCDOC1.UNLDC.PPP2' DISP(SHR)
       UTL  TEMPLATE UTLDISC DSN 'ROYC.ROYCDOC1.SDISC.PPP2'..         UNIT SYSD
       UTL  TEMPLATE UTLERR DSN 'ROYC.ROYCDOC1.SERR.PPP2'..         UNIT SYSDA.
       UTL  TEMPLATE UTLMAP DSN 'ROYC.ROYCDOC1.SMAP.PPP2'..         UNIT SYSDA
       UTL  TEMPLATE UTLUT1 DSN 'ROYC.ROYCDOC1.SUT1.PPP2'..         UNIT SYSDA
       UTL  TEMPLATE UTLOUT DSN 'ROYC.ROYCDOC1.SOUT.PPP2'..         UNIT SYSDA
       UTL  UTILFROM ROYC.ROYCDOC1.CNTLC.PPP2..         ADD(SORTNUM 8 SORTDEVT
       UTL  TEMPLATE UTLUT1 DSN 'ROYC.ROYCDOC1.SUT1.PPP2'..         UNIT SYSDA
       UTL  TEMPLATE UTLOUT DSN 'ROYC.ROYCDOC1.SOUT.PPP2'..         UNIT SYSDA
       UTL  TEMPLATE UTLERR DSN 'ROYC.ROYCDOC1.SERR.PPP2'..         UNIT SYSDA
       UTL  CHECK DATA TABLESPACE POST.POSTTS2..  ERRDDN(UTLERR) WORKDDN(UTLUT1
       UTL  RUNSTATS TABLESPACE POST.POSTTS2..  TABLE("POSTO"."PPP2")..  INDEX(
       UTL  TEMPLATE SYSCOPY DSN 'ROYC.DSN7.IC.POST.POSTTS2(+1)'..         UNIT
       UTL  COPY TABLESPACE POST.POSTTS2 COPYDDN(SYSCOPY)
       UTL  MODIFY RECOVERY TABLESPACE POST.POSTTS2 DSNUM ALL..  DELETE AGE(35)
       ADM  ENDJOB
       ADM  ENDPARALLEL
       COM  End of Compare Apply statements
 ******************************* END OF DB2 DATA
*******************************
```

Figure 312. The contents of the new WSL (2)

9. Exit to return to the **Work Statement List Library (ADB2W1)** panel.
10. Use the R line command to run the WSL.
11. In the prompt, specify a library name, a prefix to use for the job name, and whether the job name should equal the member name.

    The following figure shows the jobs that are created when you select Run Work Statement List.

```
   Menu  Functions  Utilities  Help
 ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
 EDIT     ROYC.DOCM.CNTL                                     Row 00001 of 00006
 Command ===>                                                  Scroll ===> CSR
    Name     Prompt          Size    Created           Changed            ID
 . COMPARE                    188   2002/07/16   2002/07/16 17:55:30    ROYC
 . ROYCR001                    68   2002/07/16   2002/07/16 18:33:06    ROYC
 . ROYCR002                    68   2002/07/16   2002/07/16 18:33:07    ROYC
 . ROYCU001                    64   2002/07/16   2002/07/16 18:33:04    ROYC
 . ROYCU002                    62   2002/07/16   2002/07/16 18:33:04    ROYC
 . ROYC2                       82   2002/07/16   2002/07/16 18:33:05    ROYC
   **End**
```

Figure 313. The jobs that are generated from running the WSL.

In this case, the following five jobs are generated:

- Two unload jobs (ROYCU001 and ROYCU002) are created, because two tables are changing. You can run these two unload jobs in parallel.
- Job ROYC2 performs all of the DDL tasks. You can run this job after the unload jobs have successfully completed.
- Jobs ROYCR001 and ROYCR002 reload the data. You can run these jobs in parallel.

The COMPARE job is also listed. However, this job does not need to be in the same library as the other WSL jobs.

Figure 314 on page 566 shows the R0YC2 job in detail. This job includes the following statements:

RUN PROGRAM(ADBTEP2) PLAN(ADBTEP2)
Specifies that the Db2 Admin Tool Batch Restart Program (ADBTEP2) is to be run.

LIB('DMTOOL.SADBLLIB')
Specifies the library that contains ADBTEP2. This library cannot be in the STEPLIB, because the STEPLIB must be APF-authorized to run Db2 utilities.

PARM('/WORKLIST(R0YCDOC1.2),SSID(DSN7)')
Specifies parameters. The WSL name is the first part of the WORKLIST parameter. If Db2 commands or utilities are being executed, the SSID parameter is also mandatory.

SYSIN DD *
The SYSIN DD statement provides the input to ADBTEP2. This input is called a *batch statement list*. This list contains the executable statements that are derived from the WSL.

```
****** ***************************** Top of Data ******************************

DB2 Admin: Edit generated JCL

//ROYCDOC1   JOB (ROYC,B240,090,D783),&SYSUID,
//*       RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
//     MSGCLASS=H,TIME=(2),MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//     USER=&SYSUID,REGION=8M
//*
//         CLASS=U
//*
/*JOBPARM S=SY4A
//*
//*
//**********************************************************************
//*
//* DB2 BATCH MONITOR
//*
//* DB2 ADMIN GENERATED BATCH JOB.
//*
//*************************************************************ADB2WL4**
//DB2B EXEC PGM=IKJEFT01,DYNAMNBR=100
//STEPLIB  DD DISP=SHR,DSN=DSN.DSN7.SDSNEXIT
//         DD DISP=SHR,DSN=DSN.DSN7.SDSNLOAD
//SYSEXEC  DD DISP=SHR,DSN=ADB4DEVT.EXEC
//         DD DISP=SHR,DSN=GOC2BASE.EXEC
//         DD DISP=SHR,DSN=DMTOOL.SADBEXEC
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//SYSTSIN  DD *
DSN SYSTEM(DSN7)
  RUN PROGRAM(ADBTEP2) PLAN(ADBTEP2) -
  LIB('DMTOOL.SADBLLIB') -
  PARM('/WORKLIST(ROYCDOC1.2),SSID(DSN7)')
END
//SYSIN    DD *
  DROP TABLE POSTO.PPP1;
  COMMIT;
  DROP TABLE POSTO.PPP2;
  COMMIT;
-STA DB(POST) SPACE(POSTTS1);
-STA DB(POST) SPACE(POSTTS2);
  CREATE TABLE POSTO.PPP1
     (EMP                  CHAR(6) FOR SBCS DATA WITH DEFAULT NULL ,
      PROJ                 CHAR(3) FOR SBCS DATA WITH DEFAULT NULL )
    IN POST.POSTTS1
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC;
  COMMIT;
  CREATE TABLE POSTO.PPP2
     (EMP                  CHAR(6) FOR SBCS DATA WITH DEFAULT NULL ,
      DEPT                 CHAR(3) FOR SBCS DATA WITH DEFAULT NULL )
    IN POST.POSTTS2
    AUDIT NONE
    DATA CAPTURE NONE
    CCSID      EBCDIC;
  COMMIT;

...
```

Figure 314. The resulting job: ROYC2

```
...

  CREATE INDEX POSTO.PPP1X
    ON POSTO.PPP1
     (EMP                   ASC)
    USING STOGROUP SYSDEFLT
    PRIQTY 12 SECQTY 12
    ERASE  NO
    FREEPAGE 0 PCTFREE 10
    GBPCACHE CHANGED
    BUFFERPOOL BP1
    CLOSE YES
    COPY NO
    PIECESIZE 2 G;
  COMMIT;
  CREATE INDEX POSTO.PPP2X
    ON POSTO.PPP2
     (EMP                   ASC)
    USING STOGROUP SYSDEFLT
    PRIQTY 12 SECQTY 12
    ERASE  NO
    FREEPAGE 0 PCTFREE 10
    GBPCACHE CHANGED
    BUFFERPOOL BP1
    CLOSE YES
    COPY NO
    PIECESIZE 2 G;
  COMMIT;
/*
```

Figure 315. The resulting job: ROYC2 (2)

Running WSLs with the utility template for LOBs

You can run work statement lists (WSLs) with LOBs by using the utility template for LOBs or a customization skeleton, or you can run WSLs by default.

If you use the utility template for LOBs, the Run WSL function (like other functions such as ALT and MIG) will add an ADM statement (ADMIN LOBTEMPLATE) to indicate the existence of a LOB column or columns in the table or tablespace that is involved in the next UNLOAD statement.

The LOBTEMPLATE statement format is

```
ADMIN LOBTEMPLATE  <n> DSN <DSNPrefix>...<and other attributes like UNIT, SPACE...>
```

where

<n>
Indicates the existence of *n* number of LOB columns in the next unload.

<DSNPrefix>
The dataset prefix, which can have a maximum length of 35 bytes.

When the Run WSL function reads each ADMIN LOBTEMPLATE statement, the Run WSL function performs the following steps:

1. Generates a unique name for the template.

   For example, the following name: ADBL<nnnn>

   where

   ADB
   Indicates that it is an admin template.

   L
   Indicates that it is a LOB template.

   nnnn
   Is a running sequence number for each LOB template.

2. Multiplies the given template statement into *n* templates by adding a name for the template and adding a suffix for the data set, as shown in the following example:

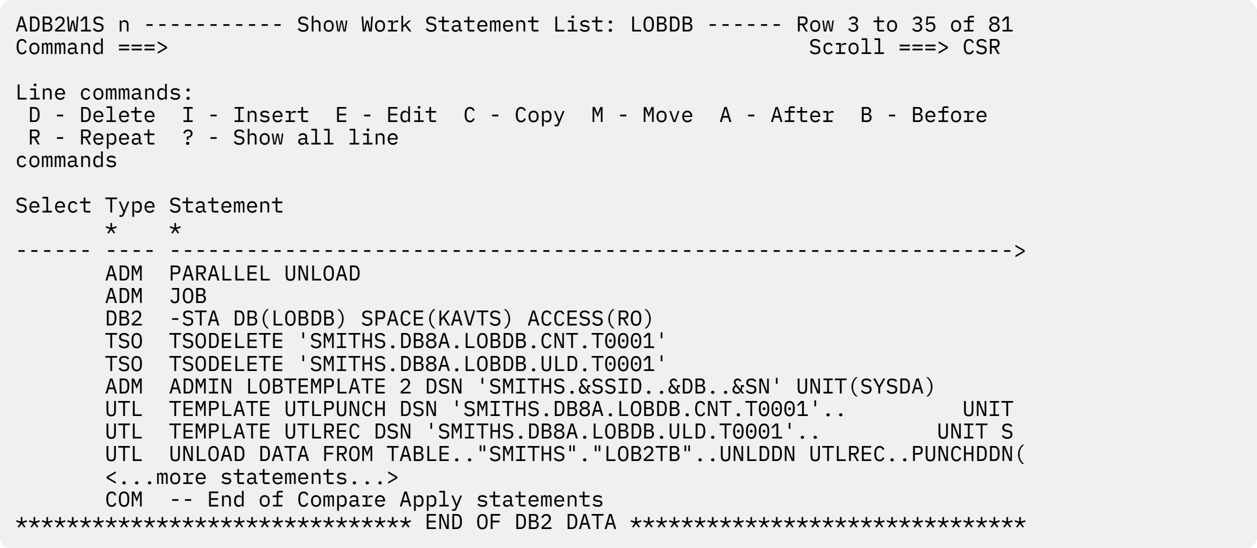
   ```
   ADMIN LOBTEMPLATE  <n> DSN <DSNPrefix>...<and other attributes like UNIT,
   SPACE...>
   ```

   The Run WSL function then replaces the preceding statement with the following set of statements:

   ```
   ADMIN LOBTEMPLATE  ADBL1 DSN <DSNPrefix>...<and other attributes like UNIT,
   SPACE...>
   ADMIN LOBTEMPLATE  ADBL2 DSN <DSNPrefix>...<and other attributes like UNIT,
   SPACE...>
   .
   .
   .

   TSODELETE ‘'SMITHS..&SSID..&DB..&SN..ADBLn'
   ADMIN LOBTEMPLATE  ADBLn DSN <and other attributes like UNIT, SPACE...>
   ```

   The Run WSL function places the templates before the corresponding UNLOAD statement by replacing the ADMIN LOBTEMPLATE statement that was generated by the Db2 Admin Tool functions.

   The ADMIN LOBTEMPLATE statement triggers ADBTEP2 to make the necessary modifications to the UNLOAD statement.



```
ADB2W1S n ----------- Show Work Statement List: LOBDB ------ Row 3 to 35 of 81
Command ===>                                                  Scroll ===> CSR

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat  ? - Show all line
commands

Select Type Statement
       *    *
------ ---- ------------------------------------------------------------------>
       ADM  PARALLEL UNLOAD
       ADM  JOB
       DB2  -STA DB(LOBDB) SPACE(KAVTS) ACCESS(RO)
       TSO  TSODELETE 'SMITHS.DB8A.LOBDB.CNT.T0001'
       TSO  TSODELETE 'SMITHS.DB8A.LOBDB.ULD.T0001'
       ADM  ADMIN LOBTEMPLATE 2 DSN 'SMITHS.&SSID..&DB..&SN' UNIT(SYSDA)
       UTL  TEMPLATE UTLPUNCH DSN 'SMITHS.DB8A.LOBDB.CNT.T0001'..         UNIT
       UTL  TEMPLATE UTLREC DSN 'SMITHS.DB8A.LOBDB.ULD.T0001'..         UNIT S
       UTL  UNLOAD DATA FROM TABLE.."SMITHS"."LOB2TB"..UNLDDN UTLREC..PUNCHDDN(
       <...more statements...>
       COM  -- End of Compare Apply statements
******************************* END OF DB2 DATA *******************************
```

Figure 316. Show Work Statement List: LOBDB (ADB2W1S)

Running WSLs with the utility template for unloading XML data

You can run work statement lists (WSLs) with XML by using the utility template for XML or a customization skeleton, or you can run WSLs by default.

If you use the utility template for XML, the Run WSL function will repeat the ADMIN XMLTEMPLATE *n* statement *n* times.

The XMLTEMPLATE statement format is

```
ADMIN XMLTEMPLATE  <n> DSN <DSNPrefix>...<and other attributes like UNIT, SPACE...>
```

where

<n>
Indicates the existence of *n* number of XML columns in the next unload.

<DSNPrefix>
The dataset prefix, which can have a maximum length of 35 bytes.

When the Run WSL function reads each ADMIN XMLTEMPLATE statement, the Run WSL function performs the following steps:

1. Appends a qualifier as needed for the template. Ensure that your data set is unique after the qualifier is appended.

   For example, the following name: ADBX<nnnn>

   where

   ADB
   Indicates that it is an admin template.

   X
   Indicates that it is an XML template.

   nnnn
   Is a running sequence number for each XML template.

2. Repeats the given template statement into *n* templates by adding a name for the template and adding a suffix for the data set, as shown in the following example:

```
ADMIN XMLTEMPLATE  <n> DSN <DSNPrefix>...<and other attributes like UNIT,
SPACE...>
```

   The Run WSL function then replaces the preceding statement with the following set of statements:

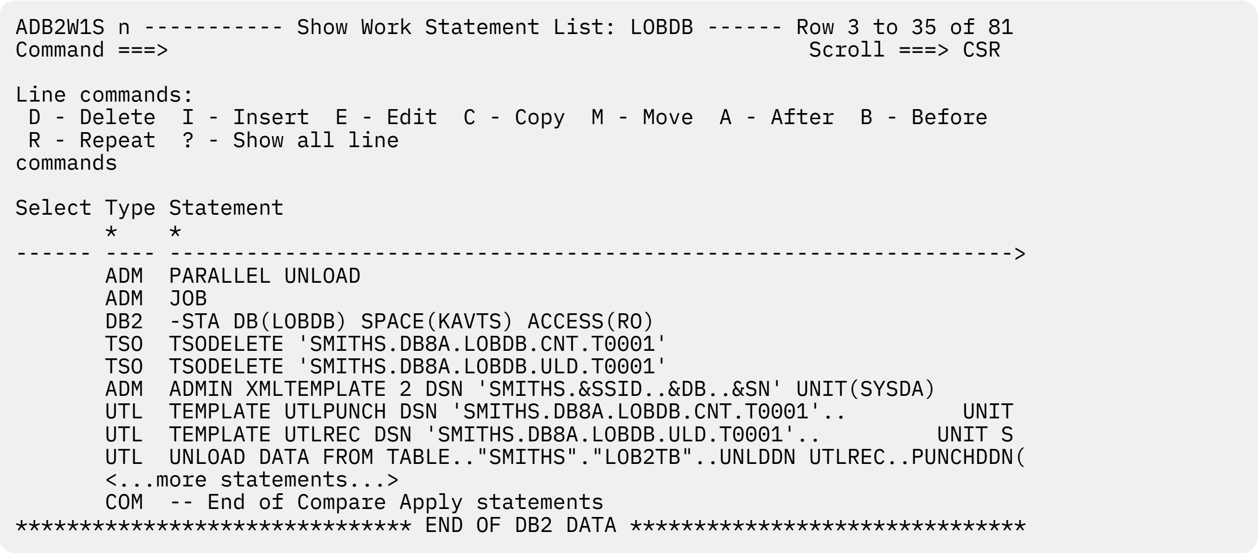
```
ADMIN XMLTEMPLATE  ADBX1 DSN <DSNPrefix>...<and other attributes like UNIT,
SPACE...>
ADMIN XMLTEMPLATE  ADBX2 DSN <DSNPrefix>...<and other attributes like UNIT,
SPACE...>
.
.
.

ADMIN XMLTEMPLATE  ADBXn DSN <and other attributes like UNIT, SPACE...>
```

   The Run WSL function places the templates before the corresponding UNLOAD statement by replacing the ADMIN XMLTEMPLATE statement that was generated by the Db2 Admin Tool functions.

   The ADMIN XMLTEMPLATE statement triggers ADBTEP2 to make the necessary modifications to the UNLOAD statement.

   Attention: The data set name pattern will be modified to include an additional qualifier when multiple XML or LOB columns exist in the object being unloaded and &TS or &SN are not included and the unload method chosen is DB2. If the unload method chosen is HPU, this check or modification is not performed as HPU will detect a data set collision and fail the unload.

   Restriction: If ADBTEP2 encounters too few XML templates for the object being unloaded, it will issue message ADB5224E and end processing.



```
ADB2W1S n ----------- Show Work Statement List: LOBDB ------ Row 3 to 35 of 81
Command ===>                                                  Scroll ===> CSR

Line commands:
 D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
 R - Repeat  ? - Show all line
commands

Select Type Statement
       *    *
------ ---- ------------------------------------------------------------------->
       ADM  PARALLEL UNLOAD
       ADM  JOB
       DB2  -STA DB(LOBDB) SPACE(KAVTS) ACCESS(RO)
       TSO  TSODELETE 'SMITHS.DB8A.LOBDB.CNT.T0001'
       TSO  TSODELETE 'SMITHS.DB8A.LOBDB.ULD.T0001'
       ADM  ADMIN XMLTEMPLATE 2 DSN 'SMITHS.&SSID..&DB..&SN' UNIT(SYSDA)
       UTL  TEMPLATE UTLPUNCH DSN 'SMITHS.DB8A.LOBDB.CNT.T0001'..         UNIT
       UTL  TEMPLATE UTLREC DSN 'SMITHS.DB8A.LOBDB.ULD.T0001'..         UNIT S
       UTL  UNLOAD DATA FROM TABLE.."SMITHS"."LOB2TB"..UNLDDN UTLREC..PUNCHDDN(
       <...more statements...>
       COM  -- End of Compare Apply statements
******************************* END OF DB2 DATA ********************************
```

Figure 317. Show Work Statement List: XMLDB (ADB2W1S)

Db2 High Performance Unload within a work statement list

When you use the Db2 Admin Tool Alter ALT and Migrate functions, you can use Db2 High Performance Unload (DB2 HPU) within a work statement list.

In addition, when using ALTER table space redefine against a single table space, you can use Db2 HPU as the unload method.

Invoking Db2 High Performance Unload within a work statement list

Before you use Db2 High Performance Unload (HPU) within a work statement list (WSL), enable HPU. The main HPU program (INZUTILB) needs to be authorized in the IKJTSOnn member of PARMLIB.

The Migrate function has a slightly different implementation than other functions, as the unload is performed before the WSL is created by using regular JCL and not under the control of the ADBTEP2 program.

For functions other than MIGRATE, you decide to use HPU when you run the work statement list. On the **Work Statement List Library (ADB2W1)** panel, enter the R line command to display the **HPU Unload Prompt (ADB2WHPU)** panel. At that time, you can decide whether to use HPU.

Restriction: The following restrictions apply to using HPU:

- If an object to be unloaded in the work statement has a security label column because the unload will fail, do not specify HPU.
- If the WSL includes an UNLOAD statement and a template substitution variable is part of the unload SYSREC template, HPU cannot be used. The Db2 UNLOAD utility is used instead, and the **HPU Unload Prompt (ADB2WHPU)** panel is not displayed.

Because using HPU is determined at run time, all WSLs are created using either UNLOAD or REORG UNLOAD EXTERNAL. You can select options R or U as the unload method when creating the work statement list. Selecting the H option does not specify that HPU will be used, but you can specify that you want to use HPU on the **HPU Unload Prompt (ADB2WHPU)** panel from the **Work Statement List Library (ADB2W1)** panel.

You can port a WSL from subsystem to subsystem. For example, if a WSL is created on a subsystem that does not have HPU enabled, you can copy that WSL to another subsystem that has HPU enabled.

If you do not select HPU at run time, the WSL runs using the Db2 UNLOAD utility. Prior to submitting the WSL jobs, you can choose between the Db2 utility and HPU.

Restriction: After the run is started, the unload method cannot be changed. For example, a job that fails using the Db2 UNLOAD utility cannot be restarted by using HPU if an object to be unloaded in the work statement has a security label column because the unload will fail.

Although Db2 Admin Tool does not run HPU in z/OS storage key 7, this situation does not cause any problems in running HPU. If you receive warning message INZU241I from HPU, you can ignore this message unless abends occur.

When a Db2 High Performance Unload job is being run using a work statement list, partitioned table spaces are unloaded by partition. The subsequent loading of the data is performed in parallel when possible; otherwise, the data sets are concatenated to form a single input stream.

Loads are performed serially in the following cases:

- When a table is loaded into a nonpartitioned table space
- When the number of partitions has changed
- When the partition key ranges have changed
- When an identity column appears in the partitioning index

Using Db2 High Performance Unload with MIGRATE and work statement lists

When migrating Db2 data, the **Migrate Parameters** panel (ADB28M) offers the option to specify a Db2 High Performance Unload unload.

You can specify that you want to unload the partitions in parallel. This option is ignored if you do not choose the HPU option. The JCL that is generated directly invokes Db2 High Performance Unload to complete the unload, as well as to create the work statement list. Because the work statement list does not contain an unload statement, no prompt is offered that asks whether Db2 High Performance Unload is required at run time. When the work statement list is run, the ADBTEP2 program automatically determines if the data was unloaded by partition and completes the appropriate steps to reload the data accordingly.

Important: You must set the parameter ULACCTRL=YES in the Db2 HPU PARMLIB, or the Db2 High Performance Unload job will not run correctly.

Db2 High Performance Unload settings

For the DB2 High Performance Unload job to run correctly, you must have the following parameter settings in the Db2 High Performance Unload PARMLIB:

- ULACCTRL=YES
- ULOPTNS=INSTREAM_XML_AS_CLOB(NO)

Using Db2 High Performance Unload in a work statement list that is not created by Db2 Object Comparison Tool, ALTER, or ALT

If Db2 High Performance Unload is enabled, all work statement lists that contain an UNLOAD or REORG UNLOAD EXTERNAL statement display the **HPU Unload Prompt** pop-up panel (ADB2WHPU) at run time.

The Db2 High Performance Unload support in Db2 Admin Tool is primarily intended to be used for a work statement list that is created by one of the Db2 Admin Tool or Db2 Object Comparison Tool functions. However, if Db2 HPU is selected at run time, any eligible unload is converted to run as a Db2 High Performance Unload unload. To be considered as an eligible unload, all of the following statements must be true:

- The UNLOAD statement, whether it be UNLOAD or REORG UNLOAD EXTERNAL, must have exactly one FROM TABLE clause, with no other keywords from the utilities FROM-TABLE-*spec*.
- The UNLOAD data set name must not exceed 38 characters. This restriction enables a suffix to be appended to the data set name that indicates the partition number.
- The DDNAME that is associated with the UNLOAD data set must be SYSREC.

Restriction: Do not code Db2 High Performance Unload syntax directly in a work statement list. Use only the Db2 utility format. When the ADBTEP2 program runs Db2 High Performance Unload on a partitioned table space, it always unloads each partition into a separate data set. For a work statement list that is not created using ALTER or Db2 Object Comparison Tool, you must ensure that subsequent handling of the output from the unload operation is managed appropriately.

How Db2 High Performance Unload reads the Db2 catalog

Db2 High Performance Unload can directly access the Db2 catalog.

Db2 Admin Tool does not specify the options that apply to non-externalized updates to the catalog data in the Db2 buffer pools. You can provide this access by defining a default in the Db2 HPU PARMLIB member using one of the following options:

- Quiesce the catalog using option QUIESCECAT=YES
- Provide direct access without flushing the Db2 buffers using QUIESCECAT=NO. This can lead to failures.
- Specify that DB2 High Performance Unload uses Db2 to perform the catalog access using option SQLACCES=YES.

Recommendation: Whenever possible, use the last option listed in the previous list. This option was provided in APAR PQ68392.

The Batch Restart programs: ADBTEP2 and ADBTEPA

The Batch Restart program, ADBTEP2, and the Authorization Switching program, ADBTEPA, are used with work statement lists and the Alter and Migrate Db2 data functions.

The Batch Restart program (ADBTEP2) enables you to restart or resume the execution of an input stream at an intermediate point, in the event that any one of the statements in that stream should fail. The Authorization Switching program (ADBTEPA) allows user IDs that are not authorized to certain objects to re-create those objects if they are implicitly dropped. ADBTEPA is used only if the auth-switching function is enabled.

Introduction to ADBTEP2

The Batch Restart program (ADBTEP2) enables you to restart or resume the execution of an input stream at an intermediate point, in the event that any one of the statements in that stream should fail.

The process involves creating or updating a record in a checkpoint table each time that a COMMIT statement is encountered in the input stream. By using this table, execution can be resumed with the first statement following the last successful commit point before the failure, bypassing all prior successfully executed statements. Of course, before restarting after a failure, you must correct the condition that caused the failure.

In comparison, ADBTEP2 does not include all functions available in Db2 Admin Tool Space Manager. For example, ADBTEP2 can support the changing of VCAT names for a table space or an index only when the VCAT names are defined within the same catalog structure.

Input types

ADBTEP2 can run the following elements from an input stream (SYSIN):

- SQL statements
- Db2 utilities
- Db2 commands
- DSN commands (including RUN)
- REXX EXECS or CLISTS

This input stream is referred to as a *batch statement list*.

Checkpoint table

ADBTEP2 is generally used in jobs that are generated by Db2 Admin Tool, but it can also be used independently. The checkpoint table is a shared resource; it is named ADBCHKPT and is stored in the *checkpoint database*. You can determine the qualifier of this table by using the ADBTEP2 package associated with the plan that you are running (ADBTEP2 by default). ADBTEP2 adds and maintains a row in the checkpoint table. This row in the checkpoint table is referenced by a worklist name parameter that is supplied to ADBTEP2. The worklist name parameter is used in conjunction with the user ID of the submitter (to ensure uniqueness). The worklist name parameter is created when the JCL is generated by Db2 Admin Tool functions and uses the work statement list name concatenated with an optional suffix.

The checkpoint table is updated at commit points to enable restarting. ADBTEP2 always performs implicit commits before and after performing functions other than SQL (for example, a Db2 utility). To issue a commit between SQL statements, add an SQL COMMIT statement. You can also instruct ADBTEP2 to commit after every statement by using the `commit_all` ADBOPT parameter.

Parameters passed to the ADBTEP2 program

When Db2 Admin Tool generates the JCL to run ADBTEP2, parameters are generated automatically and passed to ADBTEP2.

Parameters that are passed in the PARMS field of the Db2 RUN statement

The following parameters are generated and passed to ADBTEP2 in the PARMS field of the Db2 RUN statement (See RUN (DSN) (Db2 12 for z/OS)):

MAXE(*number*)

Specifies the number of DSN commands that can fail before the batch job is terminated. *number* can have one of the following values:

-1

All errors are ignored. The batch job does not stop for any error.

0

No errors are allowed. The batch job stops on the first error. 0 is the default value.

1-99

The specified number of errors are ignored. The batch job stops on the next DSN command that fails. For example, if you specify 5, the batch job stops when the sixth DSN command fails.

Any failing DSN commands that are ignored are written to the ADBHOLD table. When the job ends, if any DSN commands failed, the restart action field in the checkpoint table contains an 'H' to indicate that the table contains held records. If RESTART(YES) is specified and the batch job ended with a return code of 0, the held records are reprocessed; otherwise, the job is restarted from the last recorded commit point. If RESTART(NO) is specified, the held records are purged and the job is restarted from the beginning.

You can specify the MAXE parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Maxerrors** field.

RESTART (*restart-value*)

Specifies restart behavior. If you specify RESTART (with any value), you must also use the WORKLIST parameter.

restart-value can have one of the following values:

NO

Indicates that the job is not to be restarted, and execution starts with the first command. ADBTEP2 updates the checkpoint table, and the job can subsequently be restarted by using RESTART(YES).

YES

Indicates that the job is to be restarted from the last recorded commit point prior to a failure. RESTART(YES) is the default.

When execution begins, ADBTEP2 searches for a checkpoint record in the checkpoint table to determine where to reposition itself within the input. (The program skips committed commands.) When you specify RESTART(YES), ADBTEP2 does a basic check to ensure that the last command type that is held in the checkpoint record matches the command type that is to be attempted at restart. This check prevents an accidental reuse of a checkpoint against a different WSL.

Recommendation: Use caution when editing the input stream between ADBTEP2 failures. If the checkpoint record is not found, ADBTEP2 starts with the first command in the input stream.

FORCE

Indicates that the job is to be restarted from the last recorded commit point prior to a failure; however, ADBTEP2 does not check that the last command type that is held in the checkpoint record matches the command type that is to be attempted at restart. Because this basic check is not done, the restart point might be unintended and the results might be unpredictable. This check is done if you specify RESTART(YES).

Additionally, the check for the checkpoint record is not performed if either of the following conditions are true:

- The COMMAND_RESTART column in the ADBCHKPT table has a value of 'S' upon the restart processing.
- The checkpoint dialog **`Skip-Next`** line command is used.

You can specify the RESTART parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Restart** field. If you specify `No` for ADBTEP2 restart, a RESTART(NO) parameter is generated for each ADBTEP2 job step.

WORKLIST(*extended-name*)

Specifies a unique identifier that is used in conjunction with the user ID of the submitter to provide the key for the checkpoint record.

The full format of *extended-name* is *name.suffix*, where *name* and *suffix* are each 1-8 alphanumeric characters. The separator must be a period (.). The suffix is optional, but if the suffix is omitted, the separator must also be omitted.

For jobs that Db2 Admin Tool generates, *name* is the same as the work statement list.

Examples:

```
WORKLIST(TEST1)
WORKLIST(TEST2.N00005)
```

The following parameters, which do not control restart functions, are also passed to ADBTEP2 in the PARMS field of the Db2 RUN statement.

ALIGN(*align-value*)

Specifies how the program is to be aligned. *align-value* can have one of the following values:

MID

Aligns output from the program to the center of the page. MID is the default value.

LHS

Aligns output from the program to the left-hand side of the page.

MIXED
NOMIXED

Specifies whether the input stream can contain single-byte character set (SBCS) data and double-byte character set (DBCS) data.

MIXED

Indicates that the input stream can contain a mixture of characters from a SBCS and a DBCS.

NOMIXED

Indicates that the input stream can contain only SBCS data. NOMIXED is the default.

PCACT(*action*)

Specifies the action to take when the job is to recover a change that was made through Change Management and pending changes exist for the same objects or related objects. *action* can have one of the following values:

CANCEL

Do not run the recover job.

SUPERSEDE

Run the recover job. The recover change supersedes the pending changes, and the pending changes are set to DEFINED status.

SQLTERM(*c*)

Specifies the character that terminates an SQL statement. *c* is the character. The default SQL terminator is the semicolon (;).

SSID(*name*)

A subsystem or group attachment name to use for running non-SQL commands or functions. This name should be the same as the name that is specified in the DSN SYSTEM command before the RUN command that invokes ADBTEP2. The SSID parameter is required if any non-SQL Db2 function is included in the input stream, such as a DSN command.

Parameters passed under the ADBTEPIN DD name

The following parameters are generated automatically and passed to ADBTEP2 in a data set with a DD name of ADBTEPIN:

ADVISORYAUTOREBUILD

Specifies whether ADBTEP2 initiates a REBUILD operation on an index when the object is in the ARBDP state.

YES

A REBUILD operation is attempted. However, if **Run `REORG/REBUILD = A`** is specified (on **ALTER - Build Apply Job (ADBPALT)** panel or **Generate Analyze Job (ADB2C11A)** panel) to generate an explicit REBUILD operation, ADVISORYAUTOREBUILD=NO is used. The NO value prevents an automatic REBUILD operation that duplicates the explicit REBUILD operation.

NO

A REBUILD operation is not attempted. NO is the default.

You can specify the ADVISORYAUTOREBUILD parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Advisory Auto Rebuild** field.

Tip: To prevent ADBTEP2 from scheduling any automatic REBUILD operations, you must set AUTOREBUILD, ADVISORYAUTOREBUILD and STOGROUPAUTOREOR all to NO.

Related information:

REBUILD-pending status (Db2 12 for z/OS)

ADVISORYAUTOREORG

Specifies whether ADBTEP2 initiates a REORG operation on a table space when the object is in the AREOR or AREO* state.

YES

A REORG operation is attempted. However, if **Run `REORG/REBUILD = A`** is specified (on **ALTER - Build Apply Job (ADBPALT)** panel or **Generate Analyze Job (ADB2C11A)** panel) to generate an explicit REORG operation, ADVISORYAUTOREORG=NO is used. The NO value prevents an automatic REORG operation that duplicates the explicit REORG operation.

NO

A REORG operation is not attempted. NO is the default.

You can specify the ADVISORYAUTOREORG parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Advisory Auto Reorg** field.

Tip: To prevent ADBTEP2 from scheduling any automatic REORG operations, you must set AUTOREORG, ADVISORYAUTOREORG, and STOGROUPAUTOREOR all to NO.

Related information:

REORG-pending status (Db2 12 for z/OS)

AUTOREBUILD

Specifies whether ADBTEP2 initiates a REBUILD operation on an index when the object is in the RPDB, RPDB*, or PSRBD state.

YES

A REBUILD operation is attempted. However, if **Run `REORG/REBUILD = M`** or **Run `REORG/REBUILD = A`** is specified (on **ALTER - Build Apply Job (ADBPALT)** panel or **Generate Analyze Job (ADB2C11A)** panel) to generate an explicit REBUILD operation,AUTOREBUILD=NO is used. The NO value prevents an automatic REBUILD operation that duplicates the explicit REBUILD operation.

YES is the default.

NO

A REBUILD operation is not attempted.

You can specify the AUTOREBUILD parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Auto Rebuild** field.

Tip: To prevent ADBTEP2 from scheduling any automatic REBUILD operations, you must set AUTOREBUILD, ADVISORYAUTOREBUILD and STOGROUPAUTOREOR all to NO.

Related information:

REBUILD-pending status (Db2 12 for z/OS)

AUTOREORG

Specifies whether ADBTEP2 initiates a REORG operation on a table space when the object is in the REORP state.

YES

A REORG operation is attempted. However, if **Run `REORG/REBUILD = M`** or **Run `REORG/REBUILD = A`** is specified (on **ALTER - Build Apply Job (ADBPALT)** panel or **Generate Analyze Job (ADB2C11A)** panel) to generate an explicit REORG operation, AUTOREORG=NO is used. The NO value prevents an automatic REORG operation that duplicates the explicit REORG operation.

YES is the default.

NO

A REORG operation is not attempted.

You can specify the AUTOREORG parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Auto Reorg** field.

Tip: To prevent ADBTEP2 from scheduling any automatic REORG operations, you must set AUTOREORG, ADVISORYAUTOREORG, and STOGROUPAUTOREOR all to NO.

Related information:

REORG-pending status (Db2 12 for z/OS)

STOGROUPAUTOREOR

Specifies whether ADBTEP2 initiates a REORG or REBUILD operation after an ALTER STOGROUP statement is executed for the table space or index.

YES

A REORG or REBUILD operation is attempted. However, if **Run `REORG/REBUILD = A`** is specified (on **ALTER - Build Apply Job (ADBPALT)** panel or **Generate Analyze Job (ADB2C11A)** panel) to generate an explicit REORG or REBUILD operation, STOGROUPAUTOREOR =NO is used. The NO value prevents an automatic REORG or REBUILD operation that duplicates the explicit REORG operation.

NO

A REORG or REBUILD is not attempted. NO is the default.

You can specify the STOGROUPAUTOREOR parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Auto Reorg/Rebuild after STOGROUP change** field.

Tip: To prevent ADBTEP2 from scheduling any automatic REORG operations, you must set AUTOREORG, ADVISORYAUTOREORG, and STOGROUPAUTOREOR all to NO. To prevent ADBTEP2 from scheduling any automatic REBUILD operations, you must set AUTOREBUILD, ADVISORYAUTOREBUILD and STOGROUPAUTOREOR all to NO.

AC

Specifies whether ADBTEP2 automatically resolves CHECK-pending status.

YES

Any CHECK-pending states are automatically resolved by ADBTEP2.

ADBTEP2 tracks the following statements, utilities, and processes that can place an object in a CHECK-pending state:

- ALTER TABLE ... ADD FOREIGN KEY
- ALTER TABLE .... ADD CONSTRAINT
- LOAD REPLACE
- LOAD ENFORCE(NO)
- Recovery with the RECOVER utility to a point in time
- An auto-check prior to running the COPY utility
- An auto-check after a CHECKEND for the CHECK DATA utility
- A final auto-check at the end of the SYSIN input stream

If one of these statements or processes is encountered, ADBTEP2 runs the CHECK DATA utility to remove the CHECK-pending state.

NO

Any CHECK-pending states are not automatically resolved by ADBTEP2. NO is the default.

Restriction: Db2 Admin Tool builds the CHECK DATA statement and all CHECK parameters that are used during auto-check processing. You cannot specify any other parameters.

You can specify the AC parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **AutoCheck** field.

BINDERROR

Specifies how ADBTEP2 is to handle bind or rebind errors.

MAXE

The failing BIND or REBIND command is written to the ADBHOLD table. Whether ADBTEP2 continues to process the input stream is determined by the MAXE parameter. See "MAXE(number)" on page 573. (The failing BIND or REBIND command is counted as an error by the MAXE parameter.)

SAVE

The failing BIND or REBIND command is written to the ADBHOLD table, and ADBTEP2 continues to process the input stream.

IGNORE

The failing BIND or REBIND command is ignored. It is not written to the ADBHOLD table, and ADBTEP2 continues to process the input stream.

You can specify the BINDERROR parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **BindError** field.

PENDINGCHANGESCHECK

Specifies whether ADBTEP2 checks for Db2 pending changes before executing a DROP statement. This check avoids losing those pending changes as part of the DROP action.

YES

ADBTEP2 checks for pending changes. If any exist, the DROP statement is not executed.

NO

ADBTEP2 does not check for pending changes. The DROP statement is executed regardless of whether pending changes exist.

You can specify the PENDINGCHANGESCHECK parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Check at DROP** field.

Log DIAG

Specifies whether diagnostic messages are written to the ADBDIAG data set. IBM Software Support can use this file to determine the cause of a failure.

YES

Messages are written to ADBDIAG.

NO

Messages are not written to ADBDIAG.

You can specify this value on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Log DIAG** field.

LOAD Summary Report

Specifies whether ADBTEP2 produces a LOAD summary report.

YES

The report is produced.

NO

The report is not produced.

You can specify this value on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **LOAD Summary Report** field.

ADMUNLLOBXML

Specifies how ADBTEP2 processes UNLOAD statements when the object to be unloaded is an image copy of a table space and a table in that table space contains a LOB or XML column.

E

ADBTEP2 ends with an error.

U

The base object is unloaded instead.

You can specify ADMUNLLOBXML on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **LOB/XML IC Unload** field.

ADMUNLNOIC

Specifies how ADBTEP2 processes UNLOAD statements when the object to be unloaded is an image copy of a table space and no image copy can be found.

E

ADBTEP2 ends with an error.

U

The base object is unloaded instead.

You can specify ADMUNLNOIC on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Missing IC Unload** field.

SPANNED

Controls whether the SPANNED YES option is added to the UNLOAD statement. SPANNED YES specifies that data is be unloaded into a data set with RECFM=VBS.

YES

SPANNED YES is added to the UNLOAD statement. The SPANNED YES clause is added only if the object being unloaded has a LOB or XML column.

NO

SPANNED YES is not added to the UNLOAD statement.

You can specify SPANNED on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Spanned** field.

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

SQLFORMAT

Specifies how ADBTEP2 pre-processes SQL statements before passing them to Db2.

SQLCOMNT

ADBTEP2 does not discard SQL comments. ADBTEP2 automatically terminates each SQL comment with a line feed character (hex 25) unless the comment is already terminated by one or more line-formatting characters.

The SQLCOMNT option is suitable for all SQL, but it is intended primarily for SQL procedural language processing.

If you specify SQLCOMNT, you must add the following option manually to ADBTEPIN DD:

```
SQLFORMAT = 'SQLCOMNT'
```

TIMEOUT_RETRIES

Specifies the maximum number of times that ADBTEP2 is to retry either executing the statement or restarting from the last checkpoint when one of the following conditions occurs:

- For a timeout condition with SQLCODE -913 and reason code x'00C9008E' or a resource unavailable condition (SQLCODE -904 and reason code x'00C200EA'), ADBTEP2 retries executing the statement.
- For a timeout condition with SQLCODE -911 and reason code x'00C9008E', ADBTEP2 tries restarting from the last checkpoint.

You can specify an integer value from 0 to 99. A value of 0 means that ADBTEP2 is not to attempt any retries. 0 is the default.

You can specify the TIMEOUT_RETRIES parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Timeout Retries** field. You can also specify this parameter in the adbtep2_timeout_retries CM batch parameter.

TIMEOUT_WAIT_TIME

Specifies the duration, in seconds, between retries by ADBTEP2. You can specify an integer value from 1 to 3600. The default is 120.

You can specify the TIMEOUT_WAIT_TIME parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Timeout Wait Time** field. You can also specify this parameter in the adbtep2_timeout_wait_time CM batch parameter.

RETRY_DEPRECATED_OBJ

Specifies whether ADBTEP2 is to retry an SQL statement at a lower Db2 function level to create a deprecated object.

This parameter applies only if you are running at Db2 12 function level 504 or higher.

YES

Retry the statement.

When the current application compatibility level is Db2 12 function level 504 or higher, and an SQL statement returns SQLCODE -20008 (because it is trying to create a deprecated object), ABDTEP2 takes the following actions:

- Sets the APPLCOMPAT value to V12R1M503
- Retries the statement

- Changes APPLCOMPAT back to the previous value.

YES is the default.

NO

Do not retry the statement.

You can specify the RETRY_DEPRECATED_OBJ parameter on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Retry Deprecated Obj** field. You can also specify this parameter in the adbtep2_retry_deprecated_obj CM batch parameter.

TSACCESS

Controls whether the access state (RW/RO/UT/STOP) of the table space is preserved.

YES

At the end of ADBTEP2 execution, the table space is placed back in the same restrictive state that it was in before issuing the START DATABASE command.

YES has the following restrictions:

- Partition-level access states are not preserved.
- YES is valid for only one execution of ADBTEP2 unless you are restarting a failed job. The access state is not preserved if the job has multiple invocations of ADBTEP2.
- The access state is not preserved when an APPLY job is created.

NO

The access state is not preserved. NO is the default.

You can specify TSACCESS on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Preserve tablespace access state** field.

USE_DSNTPSMP

Specifies whether to call DSNTPSMP to create external SQL procedures from the generated DDL statements.

YES

DSNTPSMP is called.

NO

DSNTPSMP is not called. The CREATE PROCEDURE statement is executed, but you need to create the external load module and bind the package.

NO is the default value.

You can specify USE_DSNTPSMP on the **Batch Job Utility Parameters (ADB2UPA)** panel in the **Use DSNTPSMP** field.

Overriding work statement list (WSL) restart parameters

You can override the parameters that ADBTEP2 uses when performing a restart.

1. Open **Work Statement List Library (ADB2W1)** panel (Option **W** > **1** from **DB2 Administration Menu (ADB2)** panel).
2. Look for the WSL that you want to restart. The **Restart** column for this WSL should contain Y. Y indicates that checkpoint records exist for that work statement list.
3. Type the R line command next to the WSL.
4. On the **Specify Restart Information** panel, type the V line command to edit the restart information.
5. Override the parameters as needed. You can override the following parameters:

   Decfloat Rounding Mode

   Specifies the system default action that is used for rounding decimal floating point values.

   Path

   Specifies the SQL path that is used when resolving unqualified function names, procedure names, data type names, and module object names in dynamically prepared SQL statements.

Precision
Sets the CURRENT PRECISION special register.

Routine Version
Sets the CURRENT ROUTINE VERSION special register.

Rules
Sets the CURRENT RULES special register.

SCHEMA
Specifies the CURRENT SCHEMA special register to use at the restart point.

Server
Specifies the location name of the current server.

SQLID
Specifies the current SQL ID.

Use of a REXX routine with the ADBTEP2 program

A REXX routine can provide statements to ADBTEP2 for processing.

You call a REXX routine from ADBTEP2:

```
REXX %<name> [parm];
```

To provide input to ADBTEP2, you use a functional comment before the syntax. The comment informs ADBTEP2 that the REXX routine is providing information for ADBTEP2 to process. You can provide input for the following functions:

- User statements that are in a form that can be processed by ADBTEP2, for example SQL statements, Db2 commands, or DSN commands.
- Iterative processing

You end the input statements with a semicolon (;).

You must issue DSNREXX DISCONNECT in the REXX routine before you can use any command that requires ADBTEP2 to connect to Db2.

You can provide information to ADBTEP2 through the user (USERINFO) and utility information (UTILINFO) functions. You can specify a tolerance threshold for utility errors. And you can allocate output from REXX-provided statements processed by ADBTEP2 to a USRPRINT file.

User input

The user input function enables the REXX routine to provide statements on the REXX data stack to ADBTEP2.

Use the following syntax:

```
--#GET INPUT FROM STACK
REXX %<name> [parms];
```

The return code from the REXX routine specifies the action that ADBTEP2 takes:

RC=0
Statements are present on the data stack. The REXX routine writes statements onto the data stack for ADBTEP2 to process. ADBTEP2 pulls the statements from the data stack and processes statements until all statements are processed or until an error occurs.

RC=4
No statements are present on the data stack.

RC<>0, RC<>4
An error occurred and ADBTEP2 is directed to end processing.

Iterative input

The iterative input function prompts ADBTEP2 to repeat invocation of a REXX routine.

The syntax is as follows:

```
--#GET INPUT FROM STACK WITH ITERATION
REXX %<name> [parms];
```

The return code from the REXX routine specifies the action that ADBTEP2 takes:

RC=0
Statements are present on the data stack. ADBTEP2 pulls the statements from the data stack and processes statements until all statements are processed or until an error occurs.

RC=4
No statements are present on the data stack.

Until RC=4
ADBTEP2 reinvokes the REXX routine to get more statements until the REXX routine ends with RC=4.

RC<>0, RC<>4
An error occurred and ADBTEP2 is directed to end processing.

User information

The user information function enables the REXX routine to provide information for iterative REXX calls. The user information function is for iterative input only.

The syntax that prompts ADBTEP2 to process a REXX statement is as follows:

```
USERINFO <string>;
```

The user information statement enables the REXX routine to identify the work that is passed to ADBTEP2. ADBTEP2 writes the statement back to the data stack when the REXX routine is invoked the next time, and only if the call is part of iterative input processing.

The following example shows how you can call a REXX routine that passes a USERINFO string to ADBTEP2 and directs ADBTEP2 to run statistics on a tablespace:

```
/* rexx */
arg exitrc
queue "USERINFO RUN RUNSTATS ON A TABLE SPACE;
queue "RUNSTATS TABLESPACE ADBDCHG.ADBSPF1",
        "  INDEX",
        "   (",
        "    ALL",
        "   )",
        "  SHRLEVEL CHANGE;"
queue ""
exit exitrc
```

Utility Information

The utility information function enables a REXX routine to provide utility identification information, through ADBTEP2, to Db2.

The syntax that prompts ADBTEP2 to receive utility identification information from a REXX routine and to pass the information to Db2 is as follows:

```
UTILINFO [SYSTEM<ssid>,][UID=<utility-id>],[UTPROC=<utproc-string>];
```

The UTILINFO statement must precede the utility statements to which they apply. Multiple parameters must be separated by a comma. The statement must end with a semi-colon (;).

When parameters are not provided in the REXX statement, the default action is for ADBTEP2 to use parameters that are passed to ADBTEP2:

SYSTEM
: The value of the SSID() parameter that is passed to ADBTEP2 and then is passed to Db2

UID
: The value of the WORKLIST() parameter that is passed to ADBTEP2 and then is passed to Db2

UTPROC
: blank. Passes the supplied JCL procedure, if any, to Db2.

You can call a REXX routine that directs ADBTEP2 to pass Db2 utility parameters, SYSTEM and UID, to Db2. In the following example, the system name and utility ID are passed to ADBTEP2, and then ADBTEP2 runs the RUNSTATS utility:

```
/* rexx */
arg exitrc
queue "UTILINFO SYSTEM='DSNX',UID='VNDR2';"
queue "RUNSTATS TABLESPACE ADBDCHG.ADBSPF1",
        "  INDEX",
        "   (",
        "    ALL",
        "   )",
        "  SHRLEVEL CHANGE;"
queue ""
exit 0
```

Tolerance threshold for Db2 utility command error return codes

The tolerance threshold enables you to specify the error return code number, for a Db2 utility command error, to be tolerated during the processing of REXX statements. When the specified threshold is exceeded, ADBTEP2 stops processing.

The syntax that specifies the return code of errors that are tolerated is as follows:

```
--#SET TOLUTILERR n
```

The value of *n* is the return code number and must be an integer between 4 to 32767. When processing iterative statements in a REXX routine, the REXX routine, that includes Db2 utility commands, iterates until a return code that is beyond the threshold is encountered or until ADPTEP2 completes execution.

The following example shows that you specify return code tolerance before you specify a user input statement:

```
--#SET TOLUTILERR 7
--#GET        INPUT    FROM STACK WITH ITERATION
REXX T2IN2 0;
```

In the example, if the return code for a Db2 utility command error exceeds the value 7, ADPTEP2 stops processing.

User Print

The user print function enables you to send output from REXX statements processed by ADBTEP2 to a USRPRINT file.

A USRPRINT file contains output only from Db2. USRPRINT is processed only when a USRPRINT DD statement is provided.

To use USRPRINT, the following requirements must be met:

- SYSPRINT and USRPRINT must be preallocated.
- SYSPRINT must be allocated as a non-spool data set with the DISP option as MOD.
- USRPRINT must use the same data set attributes except the DISP option.

Use SYSPRINT and USRPRINT DD statements in the JCL to allocate the data sets, as shown in the following example:

```
//SYSPRINT DD DSN=<your data set>,
//            DISP=(MOD,CATLG,CATLG),
//            SPACE=(TRK,(10,10,0)),LRECL=137,RECFM=VB,BLKSIZE=141
//USRPRINT DD DSN=<your data set>,
//            DISP=(NEW,CATLG,CATLG),
//            SPACE=(TRK,(10,10,0)),LRECL=137,RECFM=VB,BLKSIZE=141,
//            VOL=SER=<volume name>
```

Data sets that ADBTEP2 uses

The ADBTEP2 program uses several data sets during its operation.

The following table lists the data sets that the ADBTEP2 program uses. The table lists the DD name that is used to identify the data set and a description of the data set. Include statements in your JCL for each required data set and any optional data sets that you want to use.

Table 31. Data sets that ADBTEP2 uses

Data set	Description
SYSIN	Input data set that contains the input stream or batch statement list, which is supplied at run time to the batch restart program.
SYSEXEC	Input data set that contains the Admin Tool EXECs
SYSTSIN	Input data set that is used to control your background job. For more details see SYSTSIN DD Statement (z/OS 3.1.0)
SYSPRINT [1]	Output data set for Db2 messages. This data set also contains ADBTEP2 messages when ADBSYSPT is not present. When the REXX user input feature is used, the data set must be allocated with the MOD as DISP option. The data set must not be a spool file, for example, USRPRINT must be defined.
ADBSYSPT[1]	Output data set for ADBTEP2 messages.
USRPRINT [1]	Output message data set for Db2 messages. This data set is used when the REXX user input feature is used. The data set must be pre-allocated. See "User Print" on page 583.
SYSTSPRT	Output data set for your background job. For more details, see SYSTSPRT DD Statement (z/OS 3.1.0).
MSGLIB	Data set that contains the IBM Language Environment® (LE) messages

Note:

1. Depending on which output data sets you allocate, Db2 messages and ADBTEP2 messages are written to these data sets as follows:

DD statements included in your JCL	Where Db2 output is written	Where ADBTEP2 output is written
SYSPRINT only	SYSPRINT	SYSPRINT
SYSPRINT and ADBSYSPT	SYSPRINT	ADBSYSPT

DD statements included in your JCL	Where Db2 output is written	Where ADBTEP2 output is written
SYSPRINT (as a data set) and USRPRINT	SYSPRINT and USRPRINT	SYSPRINT
SYSPRINT (as a data set), USRPRINT, and ADBSYSPT	SYSPRINT and USRPRINT	ADBSYSPT

Running ADBTEP2

ADBTEP2 is a batch restart program that is supplied by Db2 Admin Tool. During installation and customization of Db2 Admin Tool, Tools Customizer generates a sample job that you can use to run ADBTEP2.

Procedure

To run ADBTEP2:

1. Find the sample job in the product customization library. The product customization library name is site-specific.
2. Modify this sample job as follows:
   - Edit the job to conform to the conventions established in your installation. The names of job cards, data sets, plans, and subsystems are site-specific.
   - Provide the input data stream for execution, which is called the *batch statement list*.

     The batch statement list can be specified inline, as a sequential data set, or as a member of a partitioned data set. It should contain all of the SQL statements, Db2 commands, utility control statements, and other valid statements that you want to process in a single execution. Within this series of statements, separate logical tasks or units of work with a COMMIT statement. These COMMIT statements denote the points at which a failed execution can be restarted. Non-SQL functions have implicit commits, both before and after them.
   - Modify parameters as needed.

     Because all ADBTEP2 jobs are restartable, it is recommended that the `worklist` parameter is specified and provides a unique name. The RESTART parameter can be set to either YES or `NO` or used as default (YES), depending on whether the submission of the job is required to restart. ADBTEP2 is restartable regardless of the RESTART option. A job that is run with `RESTART(NO)`, can be resubmitted with `RESTART(YES)` in the event of a failure.

     If the `UNLOAD(HPU)` parameter is specified, Db2 Admin Tool uses the `DB2 NO` parameter when performing the HPU UNLOAD operation.
3. When you have modified the job and specified the parameters, submit the JCL for execution.

   Upon successful completion, both ADBTEP2 and ADBTEPA delete the checkpoint record.

   If the execution is unsuccessful, examine the output to determine the reason for the failure. Correct the error and resubmit the job.

Related concepts

"Introduction to ADBTEP2" on page 572
The Batch Restart program (ADBTEP2) enables you to restart or resume the execution of an input stream at an intermediate point, in the event that any one of the statements in that stream should fail.

"Parameters passed to the ADBTEP2 program" on page 573

When Db2 Admin Tool generates the JCL to run ADBTEP2, parameters are generated automatically and passed to ADBTEP2.

Dialog support for the batch job checkpoint table

To display and manage the checkpoint table (ADBCHKPT) that is associated with batch jobs that are running ADBTEP2, use the 2B–Display/Manage Batch Checkpoint Table option on the **DB2 System Administration panel (ADB2Z)**.

For each active batch job running ADBTEP2 and for jobs running ADBTEP2 that have terminated because of an error in the input stream, a record of that execution is present in the checkpoint table. Select option 1, Display Checkpoint Records, from the **Manage Batch Job Checkpoint Table** panel to see those records, terminate an active ADBTEP2 job, update or delete the record of an abnormally terminated job, or insert a new checkpoint record.

Important: A new checkpoint record is only inserted to replace one that was deleted accidentally.

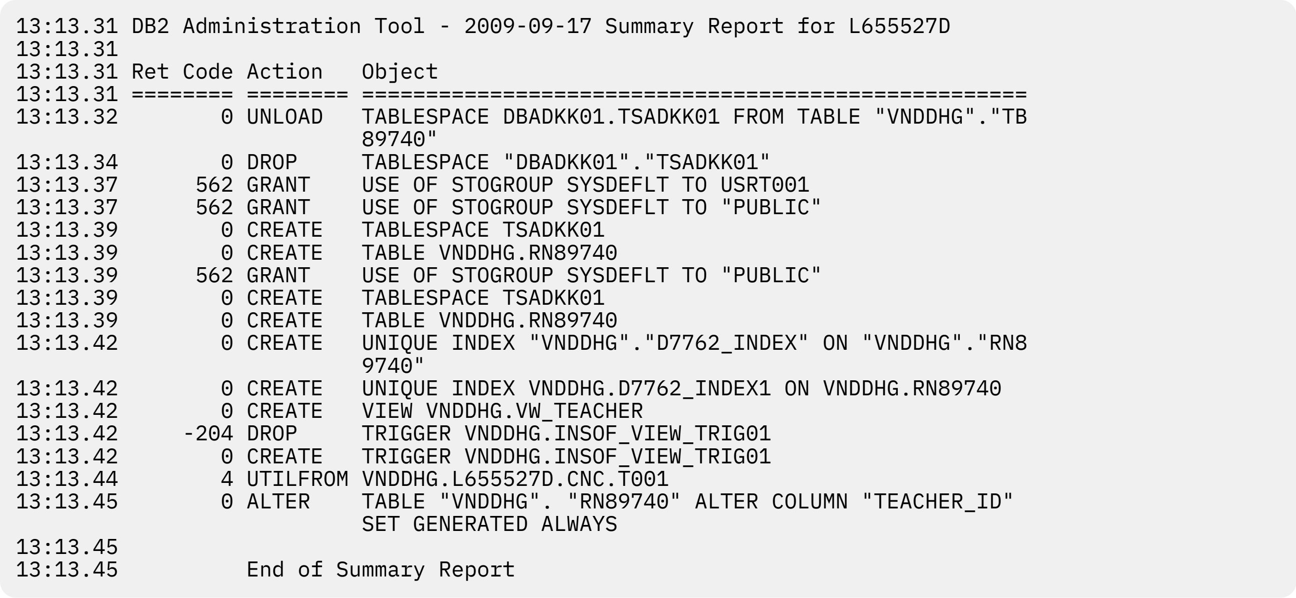
In addition, you can instruct ADBTEP2 to skip to the next commit using the N line command (skip-next).

Select option 2, Display Checkpoint Table Status, to obtain information about the checkpoint table itself, and issue any requests against the table, such as GRANT or REVOKE, that are supported by Db2 Admin Tool.

The ADBTEP2 summary report

You might want a summary report of all activity at the end of or during large or complex work statement list (WLS) runs. This report will enable you to quickly spot any object or data availability issues. The ADBTEP2 summary report appears (and grows) while any ADBTEP2 job is running, not just WSLs.

The report can be examined in SDSF, under the ADBRPTSM DD. A sample report is shown in the following figure.



```
13:13.31 DB2 Administration Tool - 2009-09-17 Summary Report for L655527D
13:13.31
13:13.31 Ret Code Action   Object
13:13.31 ======== ======== ====================================================
13:13.32        0 UNLOAD   TABLESPACE DBADKK01.TSADKK01 FROM TABLE "VNDDHG"."TB
                           89740"
13:13.34        0 DROP     TABLESPACE "DBADKK01"."TSADKK01"
13:13.37      562 GRANT    USE OF STOGROUP SYSDEFLT TO USRT001
13:13.37      562 GRANT    USE OF STOGROUP SYSDEFLT TO "PUBLIC"
13:13.39        0 CREATE   TABLESPACE TSADKK01
13:13.39        0 CREATE   TABLE VNDDHG.RN89740
13:13.39      562 GRANT    USE OF STOGROUP SYSDEFLT TO "PUBLIC"
13:13.39        0 CREATE   TABLESPACE TSADKK01
13:13.39        0 CREATE   TABLE VNDDHG.RN89740
13:13.42        0 CREATE   UNIQUE INDEX "VNDDHG"."D7762_INDEX" ON "VNDDHG"."RN8
                           9740"
13:13.42        0 CREATE   UNIQUE INDEX VNDDHG.D7762_INDEX1 ON VNDDHG.RN89740
13:13.42        0 CREATE   VIEW VNDDHG.VW_TEACHER
13:13.42     -204 DROP     TRIGGER VNDDHG.INSOF_VIEW_TRIG01
13:13.42        0 CREATE   TRIGGER VNDDHG.INSOF_VIEW_TRIG01
13:13.44        4 UTILFROM VNDDHG.L655527D.CNC.T001
13:13.45        0 ALTER    TABLE "VNDDHG". "RN89740" ALTER COLUMN "TEACHER_ID"
                           SET GENERATED ALWAYS
13:13.45
13:13.45          End of Summary Report
```

Figure 318. ADBTEP2 summary report

Restarting an ADBTEP2 job

When ADBTEP2 runs, it checks to see if a record exists within the checkpoint table that matches the `worklist` parameter for the user ID that submitted the job.

If a record does not exist, ADBTEP2 creates it and starts with the first statement in the batch statement list. If a record exists, ADBTEP2 proceeds based on the RESTART parameter. When RESTART(`NO`) is specified, ADBTEP2 starts with the first statement in the batch statement list. When either no RESTART parameter is provided or RESTART(YES) is specified, ADBTEP2 repositions itself within the batch statement list and resumes processing.

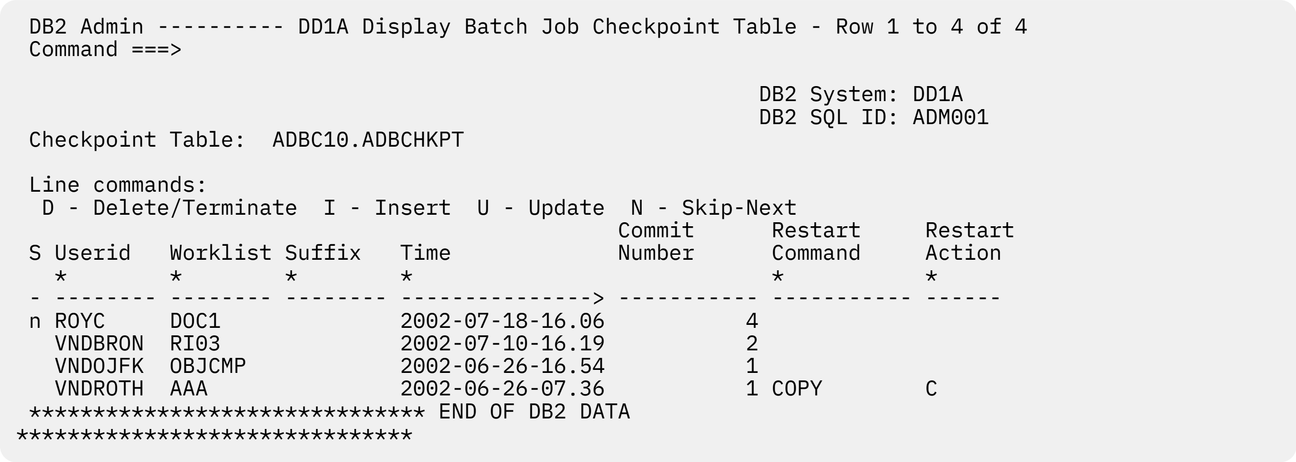
ADBTEP2 has a simple restart capability. When the failing statement is SQL, a restart occurs at the last commit point prior to the failing SQL statement, which can be either an SQL COMMIT statement or an implicit commit that is performed while successfully completing a non-SQL function, such as a Db2 command.

Tip: It is important to avoid causing ADBTEP2 to reposition incorrectly when editing the batch statement list between runs. If the only change you require is to skip to the next commit instruction, use the N (skip-next) line command instead of editing the input to ADBTEP2. For an example of using the N (skip-next) line command, see the following figure.

If the failing statement is not an SQL statement, ADBTEP2 repositions to this statement. It is possible, although not likely, for the job to fail after executing non-SQL statements and before ADBTEP2 can update and commit the checkpoint record. In this case, ADBTEP2 positions on this non-SQL statement. Non-SQL statements cannot be rolled back if a failure occurs during ADBTEP2 checkpoint/commit. If you determine that the non-SQL statement completed, you can instruct ADBTEP2 to skip this statement on restart by using the N (skip-next) line command. ADBTEP2 reports the successful implicit commits that it performs before and after non-SQL statements. You can also determine whether ADBTEP2 failed on non-SQL statements by viewing the checkpoint record: the `Restart Command` field is blank if an SQL COMMIT was the last commit or if the last commit was an implicit commit as a result of completing a non-SQL statement. If the last commit was an implicit commit ahead of non-SQL statements, the `Restart Command` field is set to the type of non-SQL statement (for example, -STA).

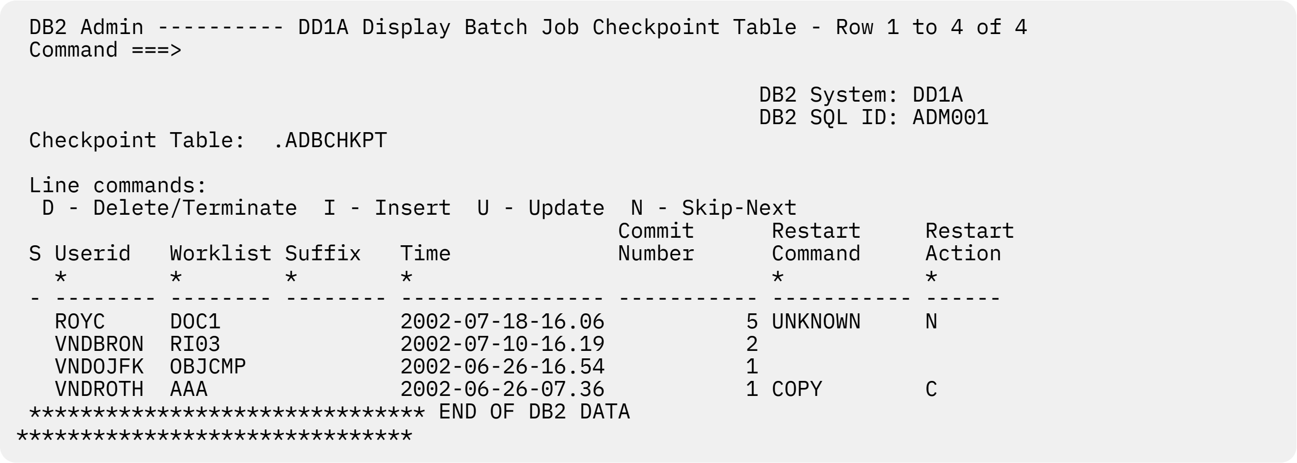
If ADBTEP2 determines that a utility was running at the time of failure, ADBTEP2 obtains information from Db2 (if the utility is known to Db2) and restarts accordingly.

The following figure illustrates the checkpoint for the job with worklist DOC1. Because the `Restart Command` field is blank, we can determine that the last instruction performed was either an SQL COMMIT or a non-SQL statement that completed with an implicit commit. If we issue an N (skip-next) line command, Figure 320 on page 588 is displayed. The checkpoint number has been increased by one.



```
 DB2 Admin ---------- DD1A Display Batch Job Checkpoint Table - Row 1 to 4 of 4
 Command ===>
                                                          DB2 System: DD1A
                                                          DB2 SQL ID: ADM001
 Checkpoint Table:  ADBC10.ADBCHKPT

 Line commands:
  D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                               Commit      Restart     Restart
 S Userid   Worklist Suffix   Time             Number      Command     Action
   *        *        *        *                            *           *
 - -------- -------- -------- ---------------> ----------- ----------- ------
 n ROYC     DOC1              2002-07-18-16.06           4
   VNDBRON  RI03              2002-07-10-16.19           2
   VNDOJFK  OBJCMP            2002-06-26-16.54           1
   VNDROTH  AAA               2002-06-26-07.36           1 COPY        C
 ******************************* END OF DB2 DATA
*******************************
```

Figure 319. Display Batch Job Checkpoint Table panel (ADB2Z2B1) – using the Skip-Next line command



```
 DB2 Admin ---------- DD1A Display Batch Job Checkpoint Table - Row 1 to 4 of 4
 Command ===>

                                                          DB2 System: DD1A
                                                          DB2 SQL ID: ADM001
 Checkpoint Table:  .ADBCHKPT

 Line commands:
  D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                               Commit      Restart     Restart
 S Userid   Worklist Suffix   Time             Number      Command     Action
   *        *        *        *                            *           *
 - -------- -------- -------- ---------------- ----------- ----------- ------
   ROYC     DOC1              2002-07-18-16.06           5 UNKNOWN     N
   VNDBRON  RI03              2002-07-10-16.19           2
   VNDOJFK  OBJCMP            2002-06-26-16.54           1
   VNDROTH  AAA               2002-06-26-07.36           1 COPY        C
 ******************************* END OF DB2 DATA
*******************************
```

Figure 320. Display Batch Job Checkpoint Table panel (ADB2Z2B1) – result of the Skip-Next line command

In Figure 321 on page 588, DOC2 has a `Restart Command` value that indicates that a COPY statement failed. The value in the `Restart Action` field determines the action to occur when ADBTEP2 repositions. For utilities, the value can be:

C
: Restart current (ADBTEP2 default)

P
: Restart phase

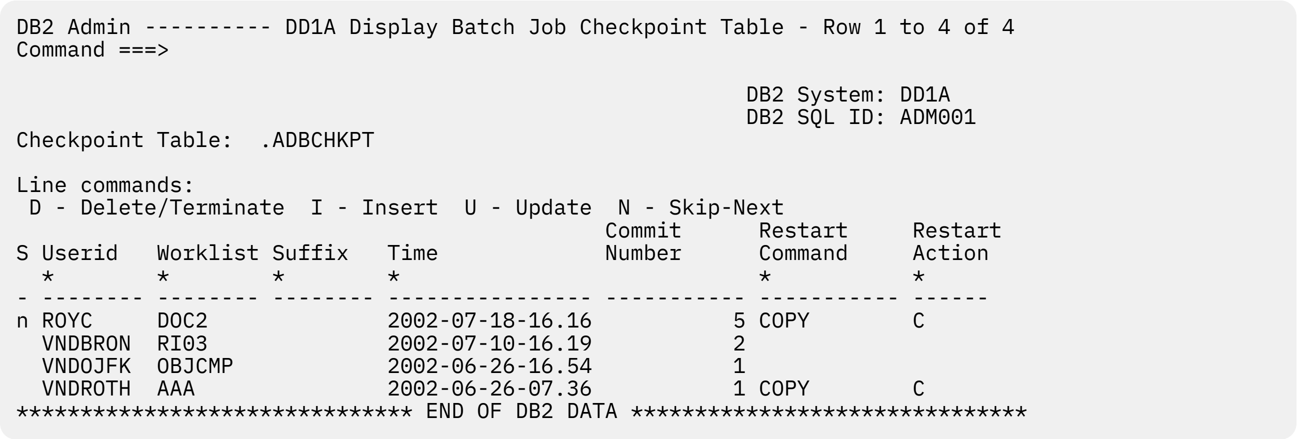
R
: Restart from the beginning of the utility

S
: Skip running the utility

The value in the `Restart Action` field can also be 'H', which indicates that the ADBHOLD table contains failed DSN commands. These failed DSN commands can be reprocessed when the job is restarted with RESTART(YES).

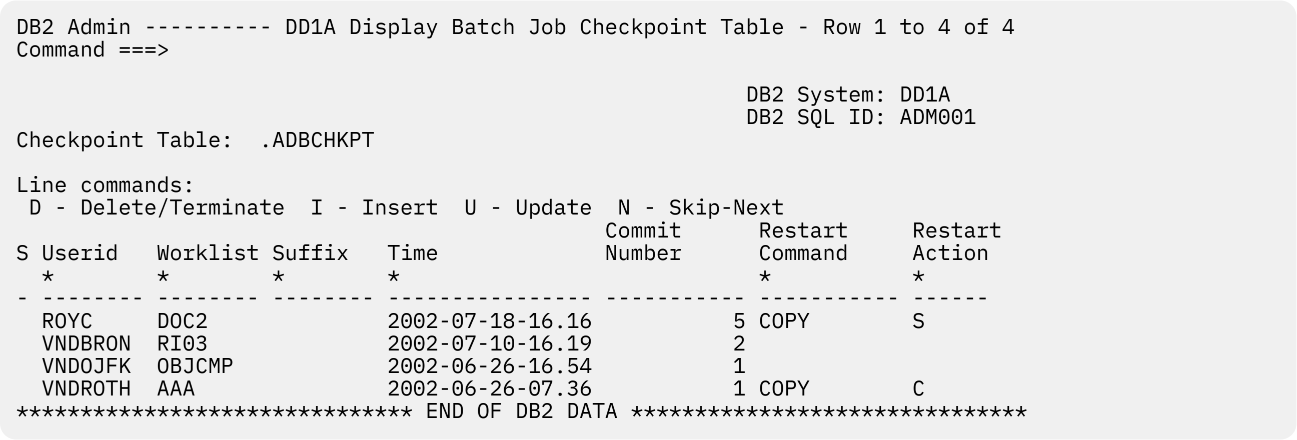
The U line command (Update) on this panel can be used to change the restart option for utilities. For example, you can change the C to an R. For non-SQL statements, only the options S (skip) and R (rerun or reissue) are valid.

Figure 322 on page 589 shows the result of using the N (skip-next) line command against DOC2. The restart command is now S and the commit number has not been increased. The `Restart Command` still displays the original type of the failing command, in this case COPY, as opposed to Figure 320 on page 588, which shows the command as UNKNOWN.



```
DB2 Admin ---------- DD1A Display Batch Job Checkpoint Table - Row 1 to 4 of 4
Command ===>

                                                         DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
Checkpoint Table:  .ADBCHKPT

Line commands:
 D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                              Commit      Restart     Restart
S Userid   Worklist Suffix   Time             Number      Command     Action
  *        *        *        *                            *           *
- -------- -------- -------- ---------------- ----------- ----------- ------
n ROYC     DOC2              2002-07-18-16.16           5 COPY        C
  VNDBRON  RI03              2002-07-10-16.19           2
  VNDOJFK  OBJCMP            2002-06-26-16.54           1
  VNDROTH  AAA               2002-06-26-07.36           1 COPY        C
******************************* END OF DB2 DATA *******************************
```

Figure 321. Display Batch Job Checkpoint Table panel (ADB2Z2B1) – reissuing the Skip-Next line command



```
DB2 Admin ---------- DD1A Display Batch Job Checkpoint Table - Row 1 to 4 of 4
Command ===>

                                                         DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
Checkpoint Table:  .ADBCHKPT

Line commands:
 D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                              Commit      Restart     Restart
S Userid   Worklist Suffix   Time             Number      Command     Action
  *        *        *        *                            *           *
- -------- -------- -------- ---------------- ----------- ----------- ------
  ROYC     DOC2              2002-07-18-16.16           5 COPY        S
  VNDBRON  RI03              2002-07-10-16.19           2
  VNDOJFK  OBJCMP            2002-06-26-16.54           1
  VNDROTH  AAA               2002-06-26-07.36           1 COPY        C
******************************* END OF DB2 DATA *******************************
```

Figure 322. Display Batch Job Checkpoint Table panel (ADB2Z2B1) – result of reissuing the Skip-Next line command

Using ADBTEP2 with LOBs

If the UNLOAD statement is preceded with a LOB template, the UNLOAD statement input is modified by ADBTEP2 before it is passed to Db2 or Db2 High Performance Unload so that ADBTEP2 can unload LOB columns.

These modifications might be obvious only by examining the job log (SDSF output). The following example is a sample job log that shows JCL that is modified by ADPTEP2.

```
****************************** Top of Data ******************************
//SMITHSD JOB (SMITHS,X,090,IE1A),'DB2 UTILITY',
//*       RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
//         REGION=0M,NOTIFY=SMITHS,
//         MSGCLASS=H,
//         CLASS=A
//*
/*JOBPARM S=SY4A
//*
//*
//**********************************************************************
//*
//* DB2 BATCH MONITOR
//*
//* DB2 ADMIN GENERATED BATCH JOB.
//*



//************************************************************ADB2WL4**
//DB2B EXEC PGM=IKJEFT01,DYNAMNBR=100
//STEPLIB  DD DISP=SHR,DSN=DSN810.SDSNEXIT
//         DD DISP=SHR,DSN=DSN810.SDSNLOAD
//MSGLIB   DD DISP=SHR,DSN=DMTOOL.SADBLLIB
//         DD DISP=SHR,DSN=DMTOOL.SGOCLLIB
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
//RNPRIN01 DD SYSOUT=*
//ADBDIAG  DD SYSOUT=*
//SYSTSIN  DD *
 DSN SYSTEM(DB8A)
 RUN  PROGRAM(ADBTEP2) PLAN(CMDBKAT) -
   LIB('DMTOOL.SADBLLIB')  -
   PARMS('/WORKLIST(TB2LOBS) SSID(DB2X)  -
 RESTART(YES)')
 END
//SYSIN    DD *
-- EDITED BY SMITHS  ON 2007/09/30 AT 02:28
-- EDITED BY SMITHS  ON 2007/09/30 AT 02:10
-- Created by SMITHS on 2007/09/30 at 02:06:58.45
TSODELETE 'SMITHS.DB2X.CNTL.LOB2DB.KAV2TS';
TEMPLATE UTLPUNCH DSN 'SMITHS.DB8A.CNTL.LOB2DB.KAV2TS'
  UNIT SYSDA;
TSODELETE 'SMITHS.DB2X.UNLD.LOB2DB.KAV2TS';
TEMPLATE UTLREC DSN 'SMITHS.DB8A.UNLD.LOB2DB.KAV2TS'
  UNIT SYSDA;
ADMIN LOBTEMPLATE ADBL1 DSN 'SMITHS.&SSID..&DB..&SN..''
UNIT SYSDA;
ADMIN LOBTEMPLATE ADBL2 DSN 'SMITHS.&SSID..&DB..&SN..'
  UNIT SYSDA;
UNLOAD TABLESPACE LOB2DB.KAV2TS
  FROM TABLE
"SMITHS"."LOB2TB"
  PUNCHDDN(UTLPUNCH)
  UNLDDN(UTLREC);
/*
```

Figure 323. Sample JCL job Log

ADBTEP2 makes the following changes (shown in bold) before passing the JCL to Db2 for processing.

1. The ADMIN LOBTEMPLATE is replaced by TEMPLATE.
2. The UNLOAD syntax is modified.

```
TSODELETE 'SMITHS.DB8A.CNTL.LOB2DB.KAV2TS';
TEMPLATE UTLPUNCH DSN 'SMITHS.DB8A.CNTL.LOB2DB.KAV2TS'
  UNIT SYSDA;
TSODELETE 'SMITHS.DB8A.UNLD.LOB2DB.KAV2TS';
TEMPLATE UTLREC DSN 'SMITHS.DB8A.UNLD.LOB2DB.KAV2TS'
UNIT SYSDA;
TEMPLATE ADBL1 DSN 'SMITHS.&SSID..&DB..&SN..'
UNIT SYSDA;
TEMPLATE ADBL2 DSN 'SMITHS.&SSID..&DB..&SN..'
  UNIT SYSDA;

UNLOAD TABLESPACE LOB2DB.KAV2TS
  FROM TABLE
"SMITHS"."LOB2TB"
 (C2REGULAR,
  C3LOBCOL VARCHAR(255) CLOBF ADBL1,
  C4LOBCOL VARCHAR(255) CLOBF ADBL2)
  PUNCHDDN(UTLPUNCH)
  UNLDDN(UTLREC);
```

Figure 324. ADBTEP changes to job

Overview of ADBTEPA

ADBTEPA is used by Db2 Admin Tool functions such as ALT(alter table columns).

ADBTEPA allows user IDs that are not authorized to certain objects to re-create those objects if they are implicitly dropped.

For example, when the owner of a table performs an alter to the table that requires dropping and re-creating the table, any views on this table are also dropped. The table owner might not have the authority to re-create some or all of the views. ADBTEPA allows the owner to re-create these views.

The ADBTEPA program receives SQL as input from SYSIN (a batch statement list) and executes it. In many respects, it is similar to ADBTEP2. For example, they both use a checkpoint table to record progress through the batch statement list. ADBTEPA and ADBTEP2 can share the same checkpoint table because the table definition is identical.

The ADBTEPA program is intended for use with the Db2 Admin Tool authorization switching function.

Once enabled, ADBTEPA is used by some functions, even if you do not request the function. ADBTEPA always allows you to perform the same tasks using SQL that you can perform under your own authorization.

Using ADBTEPA is optional; however, ADBTEPA is required when you use Db2 Admin Tool authorization switching.

Prerequisite: You must enable authorization switching on your Db2 subsystem before you can use ADBTEPA.

Deprecation notice: Authorization switching is deprecated in Db2 Admin Tool. For more information, see "Deprecated functions and functions that are no longer supported in Db2 Admin Tool 12.1" on page 64.

Using ADBTEPA

Db2 Admin Tool generates JCL for ADBTEPA when Db2 Admin Tool authorization switching is enabled.

Prerequisite: ADBTEPA is used only if the auth-switching function is enabled.

The JCL can vary slightly. A user can request an authorization switch by specifying a user ID in the authorization switch ID field on the **Alter Parameters** panel. Specifying <NONE> indicates that no Db2 Admin Tool authorization switching is requested.

The following figure illustrates an example in which Db2 Admin Tool authorization switching has not been requested, but has been enabled on the subsystem.

```
//CREAT80  EXEC PGM=ADBTEPA,DYNAMNBR=100,
//  PARM='/SSID(DSN7),WORKLIST(GO)'
//STEPLIB  DD DISP=SHR,
//            DSN=DMTOOL.SADBLINK
//         DD DISP=SHR,DSN=DSN.DSN7.SDSNEXIT
//         DD DISP=SHR,DSN=DSN.DSN7.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//ADBPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ADBOPT    DD *
PLAN=ADBTEPA
//*AUTH_SWITCH_USERID=
//SYSIN    DD DISP=SHR,DSN=ROYC.GO.DDL.CONVMERG
//*
```

Figure 325. Db2 Admin Tool authorization switching example – enabled on subsystem

ADBTEPA, unlike ADBTEP2, is executed directly and not from within DSN under IKJEFT01. Consequently, the SSID PARM is required to identify the Db2 subsystem on which to run. Similarly, the plan that ADBTEPA uses must also be supplied using the ADBOPT DDNAME. ADBTEPA uses the RRSAF attachment to access Db2.

The following figure illustrates the case where an authorization switch ID has been requested to ADBAUTHS.

```
//CREAT80  EXEC PGM=ADBTEPA,DYNAMNBR=100,
//  PARM='/SSID(DSN7),WORKLIST(GO)'
//STEPLIB  DD DISP=SHR,
//            DSN=DMTOOL.SADBLINK
//         DD DISP=SHR,DSN=DSN.DSN7.SDSNEXIT
//         DD DISP=SHR,DSN=DSN.DSN7.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//ADBPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//ADBOPT    DD *
PLAN=ADBTEPA
AUTH_SWITCH_USERID=ADBAUTHS
//SYSIN    DD DISP=SHR,DSN=ROYC.GO.DDL.CONVMERG
```

Figure 326. Db2 Admin Tool authorization switching example – authorization switch requested

In this example, the ID requested was specified using the ADBOPT DDNAME.

ADBTEPA requires that only APF-authorized libraries appear in the STEPLIB, unless ADBTEPA is placed in the link list.

When Db2 Admin Tool authorization switching is enabled, the batch statement list includes system-generated comments near the start of input and after some SQL statements. Do not remove or alter these comments.

Restarting ADBTEPA after a failure

ADBTEPA is restartable in the same way as ADBTEP2.

If it fails, you can change, add, or remove the ADBOPT parameter, AUTH_SWITCH_USERID=. Using AUTH_SWITCH_USERID= implicitly causes checkpoints to be taken after every statement, even across restarts.

Recommendation: Exercise caution in modifying the batch statement list after a failure. To skip the SQL statement that fails, use the Skip-Next line command within option Z.2B, as opposed to updating the checkpoint record or the batch statement list.

Automated REORG with ADBTEP2

Certain SQL statements can create pending changes, place the object in an advisory REORG-pending state, and require that the REORG utility be run to materialize the changes. To help automate the REORG operation, ADBTEP2 initiates an automated REORG, or *auto-reorg*.

Under certain circumstances, Db2 requires templates for the UNLDDN, COPYDDN, PUNCHDDN, or DISCARDDDN data sets when performing a REORG operation. Auto-reorg uses the following default templates:

- For UNLDDDN:

  ```
  &USERID..ADBREORG.&DB..&SN..&UNIQ.
  ```

- For COPYDDN:

  ```
  &USERID..ADBCOPY.&DB..&SN..&UNIQ.
  ```

- For PUNCHDDN: If the table space is partitioned and uses relative page numbering, the following template is used:

  ```
  &USERID..ADBPUNCH.&DB..&SN..P&PA.
  ```

  Otherwise, the following template is used:

  ```
  &USERID..ADBPUNCH.&DB..&SN..&UNIQ.
  ```

- For DISCARDDN:

  ```
  &USERID..ADBDISC.&DB..&SN..&UNIQ.
  ```

You can override these templates by including the following statements at the beginning of the input stream:

```
ADMIN REORG TEMPLATE  ADBREORG <template parameters>;
```

```
ADMIN COPY TEMPLATE ADBCOPY <template parameters>;
```

```
ADMIN PUNCH TEMPLATE ADBPUNCH <template parameters>;
```

```
ADMIN DISCARD TEMPLATE ADBDISC <template parameters>;
```

The ADMIN REORG, ADMIN COPY, ADMIN PUNCH, and ADMIN DISCARD keywords are stripped from the statements.

Related reference

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

Syntax and options of the TEMPLATE control statement (Db2 12 for z/OS)

ADBOPT parameters

ADBOPT parameters are specified using the DDNAME ADBOPT.

Place the options one-per line, and always use uppercase.

ADBOPT parameters for ADBTEP2 and ADBTEPA are listed in the following table:

Table 32. ADBOPT parameters for ADBTEP2 and ADBTEPA

Parameter	Default	Usage	ADBTEP2	ADBTEPA
AUTH_SWITCH_USERID=	None	User ID to provide authority to perform SQL operations.	N/A (Return Code 12 is issued)	Optional
PLAN=	None	Plan that ADBTEPA is to use.	N/A (Ignored)	Mandatory
COMMIT_ALL=	N	Commit/ checkpoint mode: **Y** commits after every statement. **N** commits before and after non-SQL, or COMMIT statements. After setting this option to **Y**, it persists across restarts.	Optional	Optional (Ignored if AUTH_SWITCH_USERID= is specified)
ADB2UTIL=	ADB2UTIL	Allows alternative name for program ADB2UTIL	Optional	N/A (Ignored)

Pausing ADBTEP2 and ADBTEPA

You can use the ADBPAUSE statement to pause the ADBTEP2 and ADBTEPA programs at a certain point.

To restart ADBTEP2 or ADBTEPA after an ADBPAUSE statement, submit the program again with the RESTART(YES) parameter (either explicitly or by default). The program restarts at the statement that immediately follows the ADBPAUSE statement. If you submit the program using the RESTART(NO) parameter, processing starts at the first statement in the batch statement list.

Running Db2 utilities from Db2 Admin Tool

You can run Db2 utilities from Db2 Admin Tool by using the UTIL primary command or the UT, UTL, or UTIL line commands on a system catalog panel.

Running Db2 utilities on table spaces

Many Db2 utilities run against table spaces. You can use Db2 Admin Tool to specify the utility options and generate the JCL to run these utilities.

About this task

If you want to run a utility against a LISTDEF list of table spaces, see "Running utilities on LISTDEF lists" on page 609.

If you want to run redirected recovery, see "Running a redirected recovery on a table space" on page 598.

Procedure

To run Db2 utilities on table spaces:

1. Select the table spaces on which you want to run the utility:

   a) On the **DB2 Administration Menu (ADB2)** panel, select option 1.

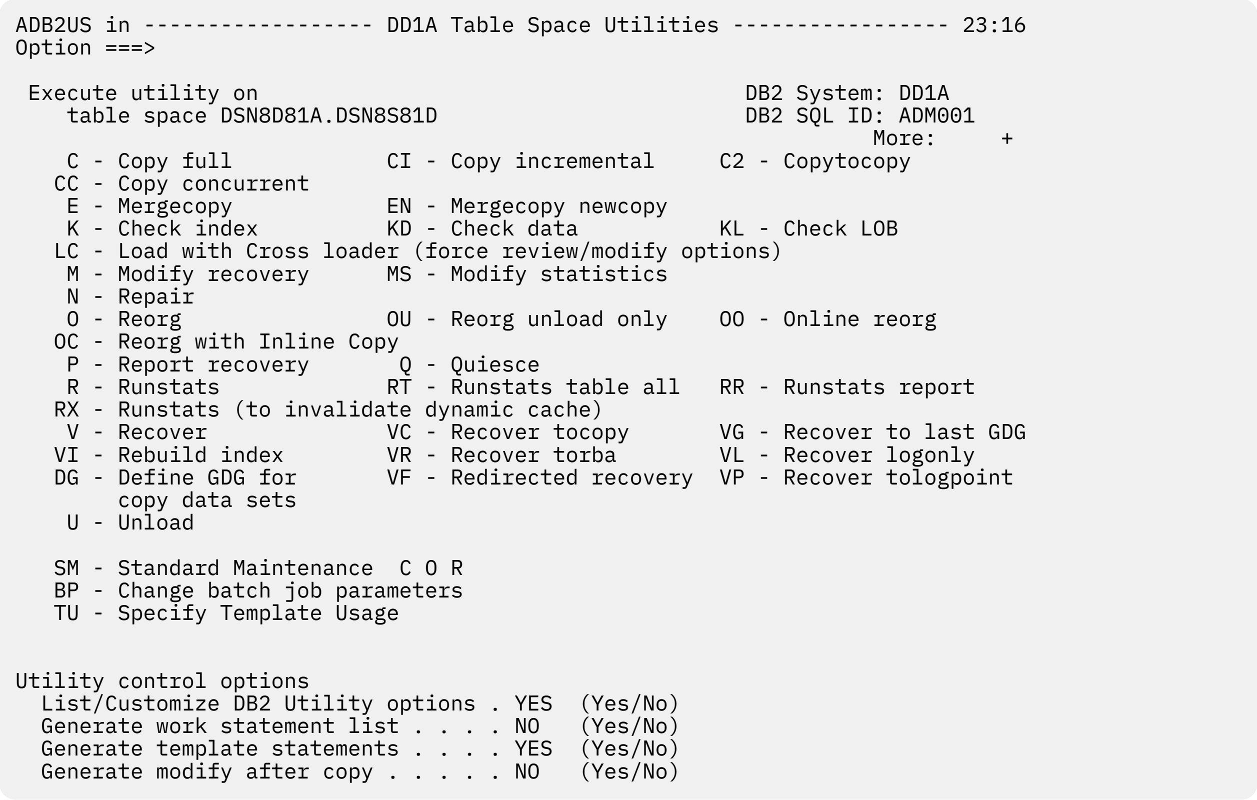
   b) On the **System Catalog (ADB21)** panel, if you want to select all table spaces in one or more databases, select option D. Otherwise, select option S. Optionally specify any filtering criteria at the bottom of the panel, and press Enter.

   c) If the **Databases (ADB21D)** panel is displayed, filter the list as needed, and specify the UTIL primary command.

   d) If the **Table Spaces (ADB21S)** panel is displayed, select one or more table spaces:

      - If you want to select one table space, issue the UT line command next to the table space name, and press Enter.
      - If you want to select more than one table space, filter the list as needed, specify the UTIL primary command, and press Enter.

   The **Table Space Utilities (ADB2US)** panel is displayed:



```
ADB2US in ------------------ DD1A Table Space Utilities ----------------- 23:16
Option ===>

 Execute utility on                                      DB2 System: DD1A
    table space DSN8D81A.DSN8S81D                        DB2 SQL ID: ADM001
                                                                   More:     +
    C - Copy full            CI - Copy incremental     C2 - Copytocopy
   CC - Copy concurrent
    E - Mergecopy            EN - Mergecopy newcopy
    K - Check index          KD - Check data           KL - Check LOB
   LC - Load with Cross loader (force review/modify options)
    M - Modify recovery      MS - Modify statistics
    N - Repair
    O - Reorg                OU - Reorg unload only    OO - Online reorg
   OC - Reorg with Inline Copy
    P - Report recovery       Q - Quiesce
    R - Runstats             RT - Runstats table all   RR - Runstats report
   RX - Runstats (to invalidate dynamic cache)
    V - Recover              VC - Recover tocopy       VG - Recover to last GDG
   VI - Rebuild index        VR - Recover torba        VL - Recover logonly
   DG - Define GDG for       VF - Redirected recovery  VP - Recover tologpoint
        copy data sets
    U - Unload

   SM - Standard Maintenance  C O R
   BP - Change batch job parameters
   TU - Specify Template Usage


Utility control options
  List/Customize DB2 Utility options . YES  (Yes/No)
  Generate work statement list . . . . NO   (Yes/No)
  Generate template statements . . . . YES  (Yes/No)
  Generate modify after copy . . . . . NO   (Yes/No)
```

Note: The **LC** option is displayed only when all of the following conditions are true:

- The table does not contain XML columns.
- The panel is displayed for one table space.
- The table space contains only one table.
- The table space is not a LOB table space.
- The target table does not contain GENERATED ALWAYS columns.

Figure 327. ***Table Space Utilities (ADB2US)*** *panel*

2. Optional: Specify the following options as needed:

BP

If you plan to generate a batch utility job stream and want to change the JOB statement or other system parameters, specify **BP** and press Enter. On the resulting **Batch Job Utility Parameters (ADB2UPA)** panel, specify the options that you want to change. Then, press PF3 to save your changes and return to the **Table Space Utilities (ADB2US)** panel.

TU

If you plan to use templates and want to specify which template to use for a given utility option, specify **TU**. Then, follow the instructions in "Associating templates with data sets" on page 439.

Tip: When you run the COPY utility, by default one copy is written to the data set that is defined by the SYSCOPY DD statement. If you want more than one copy of the output, you can create and use templates for COPYDDN 1, COPYDDN2, RECOVERYDDN1, and RECOVERYDDN2.

SM

If you want to run the listed series of utilities, specify the **SM** command. The utilities are processed in the order specified in this field. For example, the following field shows that COPY (C), REORG (O), and RUNSTATS (R) will be run in that order.

```
SM - Standard Maintenance  C O R
```

If you specify SM, you cannot set **List/Customize DB2 Utility options** to YES.

3. In the **Option** field, specify a utility to run on the selected table. For example, specify 0 to run the REORG utility.
4. Change any of the following control options, and press Enter:

List/Customize DB2 Utility options

Specify whether you want to review and edit the utility options. If the value is NO, the default options for the selected utility are used.

For the REPAIR utility (option N), you must select which REPAIR function you want to run. Therefore, Db2 Admin Tool always uses a value of YES for this field and displays a subsequent option panel, **Specify Utility Options - REPAIR TABLESPACE (ADB2USN)** panel, even if you set this field to NO.

Generate work statement list

Specify whether you want the utility control statements to be added to a work statement list (WSL). If the value is NO, an executable utility job stream is generated instead.

When you specify the CHECK utility, a batch statement list, which is similar to a WSL, is generated by default, regardless of the value of the **Generate work statement list** field. The batch statement list is required as an input file to the Batch Restart program (ADBTEP2), which manages the CHECK utility function.

Generate template statements

Specify whether you want templates to be used. If the value is YES, Db2 Admin Tool uses the active templates that are defined. (If you want to change these templates, specify the **TU** option.)

Db2 Admin Tool does not generate any TSODELETE statements to delete any existing data sets for the template first. To ensure that any existing data sets are deleted, consider using one of the following techniques when you define the template:

- Specify the data set name pattern as a GDG (generation data group) where the next data set in the sequence is generated (+1), and change the other common options so that the GDGLIMIT is 1. This setup causes the data sets in the group to roll off so that only one data set exists at any one time. For example, a data set pattern name might be specified as &db..&ts..&name..ic(+1).
- Specify a DISP option of NEW, DELETE, DELETE for the data set, if appropriate.

When you specify the CHECK utility, templates are used regardless of the value of the **Generate template statements** field, because the CHECK utility function requires the use of templates. Either the default templates or the templates that you specify are used.

Generate modify after copy
Specify whether the generated JCL includes a job step to run the MODIFY utility after a full image copy.

5. If the **Specify Utility Options** panel is displayed, specify any additional options, and press Enter.

   This panel is displayed if **List/Customize DB2 Utility options** = YES or if the utility selected is N (REPAIR) or LC (Load with Cross loader).

   You must specify the options according to the Db2 syntax rules for utilities. For additional information, restrictions, and recommendations on various utility options, see Db2 online utilities (Db2 12 for z/OS).
6. **If you requested a work statement list (WSL):** On the **Specify Work Statement List (ADB2WLDA)** panel, specify the WSL information, and press Enter.

   The utility statements are added to the WSL.
7. **If you requested a batch job stream:** On the ISPF edit session, change the generated job as needed or copy it to another it to another data set.

   You can use standard ISPF editor commands to manually modify the JCL.

   Db2 Admin Tool supports unloading tables or table spaces that produce a record length that is less than 32 KB. When a table or table space with LOB objects is unloaded, the required record length might exceed 32 KB. In this case, modify the unload job or work statement list (WSL) to specify the utility parameters that allow unloading the table or table space.

   The following figure shows an example of the generated JCL. In this example, the **C - Copy full** option was chosen on the **Table Space Utilities (ADB2US)** panel.

```
-------------------------------------------------------------------------------
EDIT       ISTJE.SPFTEMP2.CNTL                             Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
000007 //*
000008 //**********************************************************************
000009 //*
000010 //* DB2 ADMIN GENERATED JOB TO RUN COPY ON SELECTED TABLESPACES
000011 //*
000012 //************************************************************ADB2USC***
000013 //*
000014 //**********************************************************************
000015 //* STEP COPY: COPY TABLESPACE DSN8D81A.DSN8S81D
000016 //************************************************************ADB2USC1**
000017 //COPY EXEC DSNUPROC,SYSTEM=DB2X,
000018 //             LIB='SYS1.DSNDB2X.SDSNLOAD',
000019 //             UID='ISTJE'
000020 //DSNUPROC.SYSCOPY DD DSN=ISTJE.DB2X.IC.DSN8D81A.DSN8S81D(+1),
000021 //             DISP=(NEW,CATLG),
000022 //             SPACE=(8192,(7,5),RLSE),
000023 //             UNIT=SYSDA
000024 //DSNUPROC.SYSIN  DD  *
000025 COPY TABLESPACE DSN8D81A.DSN8S81D DSNUM ALL FULL YES
000026 /*
000027 //**********************************************************************
000028 //* STEP MOD: MODIFY RECOVERY TABLESPACE DSN8D81A.DSN8S81D
000029 //**********************************************************************
000030 //MOD EXEC DSNUPROC,SYSTEM=DB2X,
000031 //             LIB='SYS1.DSNDB2X.SDSNLOAD',
000032 //             UID='ISTJE'
000033 //DSNUPROC.SYSIN  DD  *
000034 MODIFY RECOVERY TABLESPACE DSN8D81A.DSN8S81D DSNUM ALL
000035     DELETE AGE(35)
000036 /*
****** **************************** Bottom of Data ****************************
```

Figure 328. Edit generated JCL panel (COPY utility)

8. Submit the generated JCL job or run the specified WSL to run the utility.

Related concepts

“Batch job parameters for utility jobs” on page 605

When you run utilities, you can change some of the batch job parameters, such as the JOB card, the EXEC statement parameters, the ADBTEP2 parameters, and the space parameters.

Related tasks

"Running a WSL" on page 551
You can run a work statement list (WSL) either in batch or online.

Related information

Db2 online utilities (Db2 12 for z/OS)

Running a redirected recovery on a table space

A *redirected recovery* is when the RECOVER utility redirects the recovery of an object (the source) to another object (the target).

Procedure

To run a redirected recovery on a table space by using the RECOVER utility:

1. Navigate to the *target table space* (the table space to which you want to recover from the source table space) and specify the UT line command.
2. On the **Table Space Utilities (ADB2US)** panel, specify option VF.

   If you want to specify additional options or control options, see steps "2" on page 595 and "4" on page 596 in "Running Db2 utilities on table spaces " on page 594
3. On the **Specify SOURCE for redirected recovery (ADB2USVF)** panel, either manually specify or look up the source database and table space (in the FROM clause), and press Enter.

   To look up an object, specify a question mark (?) in the appropriate field. The lookup feature automatically fills the values on the **Specify SOURCE for redirected recovery (ADB2USVF)** panel based on the object you select. You can also look up source image copies for the TOCOPY option. This feature allows you to browse SYSIBM.SYCOPY for an image copy of the source table space.
4. Complete the usual steps for running a utility.

   Start with step "5" on page 597 in "Running Db2 utilities on table spaces " on page 594.

Related information

Redirected recovery (Db2 12 for z/OS documentation)

Running Db2 utilities on tables

You can use Db2 Admin Tool to run the LOAD, UNLOAD, and REORG UNLOAD EXTERNAL utilities on tables. Based on the options that you choose, Db2 Admin Tool generates utility statements for you. You can choose to either save these utility statements in a work statement list (WSL) or generate a batch utility job stream.

About this task

If you want to run a utility against a LISTDEF list of tables, see "Running utilities on LISTDEF lists" on page 609.

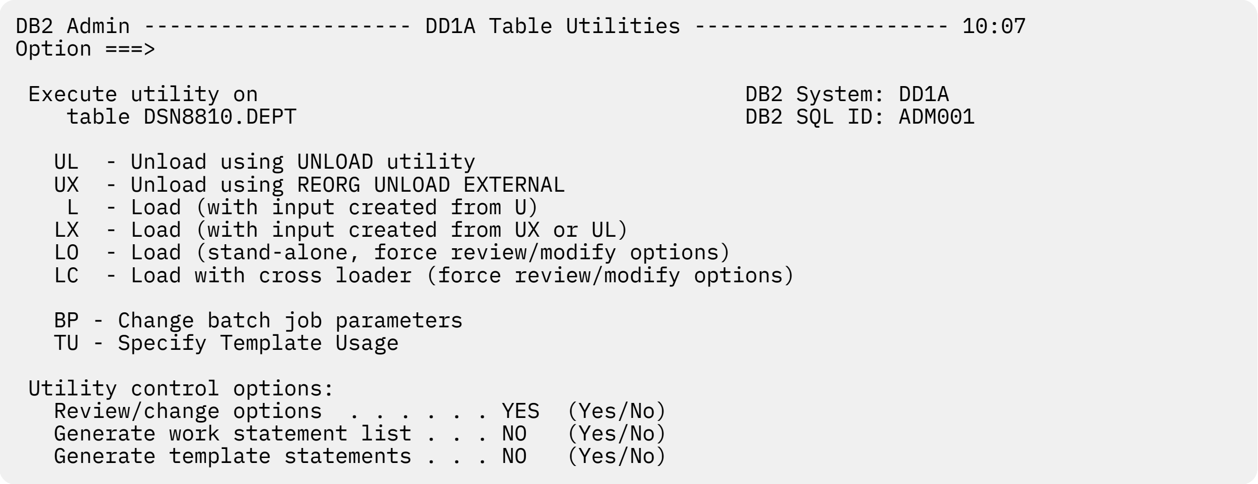
Procedure

To run Db2 utilities on tables:

1. Select the tables on which you want to run the utility:
   a) On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
   b) On the **System Catalog (ADB21)** panel, specify option T. Optionally specify any filtering criteria at the bottom of the panel, and press Enter.
   c) On the **Tables, Views, and Aliases (ADB21T)** panel, select one or more tables:

- If you want to select one table, issue the UT line command next to the table name, and press Enter.
- If you want to select more than one table, filter the list as needed, specify the UTIL primary command, and press Enter.

The **Tables Utilities (ADB2UT)** panel is displayed:



```
DB2 Admin --------------------- DD1A Table Utilities -------------------- 10:07
Option ===>

 Execute utility on                                    DB2 System: DD1A
    table DSN8810.DEPT                                 DB2 SQL ID: ADM001

   UL  - Unload using UNLOAD utility
   UX  - Unload using REORG UNLOAD EXTERNAL
    L  - Load (with input created from U)
   LX  - Load (with input created from UX or UL)
   LO  - Load (stand-alone, force review/modify options)
   LC  - Load with cross loader (force review/modify options)

   BP - Change batch job parameters
   TU - Specify Template Usage

 Utility control options:
   Review/change options  . . . . . . YES  (Yes/No)
   Generate work statement list . . . NO   (Yes/No)
   Generate template statements . . . NO   (Yes/No)
```

*Figure 329. **Tables Utilities (ADB2UT)** panel*

The **LC** option (the cross-loader function of the LOAD utility) is displayed only in the following situations:

- The table does not contain XML columns.
- The panel is displayed for only one table, not multiple tables.
- The target table does not contain GENERATED ALWAYS columns.

2. Optional: Specify the following options as needed:

   BP
   If you plan to generate a batch utility job stream and want to change the JOB statement or other system parameters, specify **BP** and press Enter. On the resulting **Batch Job Utility Parameters (ADB2UPA)** panel, specify the options that you want to change. Then, press PF3 to save your changes and return to the **Tables Utilities (ADB2UT)** panel.

   TU
   If you plan to use templates and want to specify which template to use for a given utility option, specify **TU**. Then, follow the instructions in "Associating templates with data sets" on page 439.

3. In the **Option** field, specify a utility to run on the selected table. For example, specify UL to run the UNLOAD utility.

4. Change any of the following control options, and press Enter:

   Review/change options
   Specify whether you want to review and change the utility options. If the value is NO, the default options for the selected utility are used.

   Generate work statement list
   Specify whether you want the utility control statements to be added to a work statement list (WSL). If the value is NO, an executable utility job stream is generated instead. If you specify **LO** as the utility, **Generate work statement list** must be NO.

   Generate template statements
   Specify whether you want templates to be used. If the value is YES, Db2 Admin Tool uses the active templates that are defined. (If you want to change these templates, specify the **TU** option.)

Db2 Admin Tool does not generate any TSODELETE statements to delete any existing data sets for the template first. To ensure that any existing data sets are deleted, consider using one of the following techniques when you define the template:

- Specify the data set name pattern as a GDG (generation data group) where the next data set in the sequence is generated (+1), and change the other common options so that the GDGLIMIT is 1. This setup causes the data sets in the group to roll off so that only one data set exists at any one time. For example, a data set pattern name might be specified as &db..&ts..&name..ic(+1).
- Specify a DISP option of NEW, DELETE, DELETE for the data set, if appropriate.

5. If the **Specify Utility Options** panel is displayed, specify any additional options, and press Enter.

   This panel is not displayed in the following situations:

   - If you specified **LX**
   - If you specified **Review/change options** = NO and did not specify **LO**

   You must specify the options according to the Db2 syntax rules for utilities. For additional information, restrictions, and recommendations on various utility options, see Db2 online utilities (Db2 12 for z/OS).

   Restriction: For LOAD, you cannot specify the DELIMTED option when character conversion is required.

   Example of Specify Utility Options panel for LOAD:

```
ADB2UTC n ------------- DD1A Specify Utility Options - LOAD ------------- 18:36
Command ===>

Execute utility on table ELACZ.TBTEST1
  using the following options:

Utility ID . . . . .
Unloaded data  . . .
Unloaded how?  . . . U          (U - Unload utility, R - Reorg utility)
Table/Col info . . .
PREFORMAT  . . . . .            (Yes/No)
PRESORTED  . . . . .            (Yes/No)
PRESORT  . . . . . .            (Yes/No)
RESUME . . . . . . .            (Yes/No)
  BACKOUT  . . . . . YES        (Yes/No)
SHRLEVEL . . . . . .            (N - None, R - Reference, C - Change)
REPLACE  . . . . . .            (Yes/No)
  COPYDDN1 . . . . .            (Primary copy DD name)
  COPYDDN2 . . . . .            (Backup copy DD name)
  RECOVERYDDN1 . . .            (Remote primary copy DD name)
  RECOVERYDDN2 . . .            (Remote backup copy DD name)

STATISTICS . . . . .            (Yes/No)
STATISTICS . . . . .            (Yes/No)
  TABLE schema . . .           >
        name . . . .                     > (ALL or ? for table look up)
  SAMPLE . . . . . .            (Percent to sample during RUNSTATS: 1-100)
  USE PROFILE  . . .            (Yes/No)
  COLUMN name  . . .                     > (ALL or ? for column look up)
  COLGROUP name  . .                     > (? for column look up)
    FREQVAL  . . . .            (Yes/No)
      COUNT  . . . .            (1-65535)
      OCCUR  . . . .            (M - Most, B - Both, L - Least)
    HISTOGRAM  . . .            (Yes/No)
      NUMQUANTILES .            (1-100, default 100)
  STATCLGMEMSRT  . .            (0-4096)
  INDEX ALL  . . . .            (Yes/No)
    HISTOGRAM  . . .            (Yes/No)
      NUMCOLS  . . .            (1-64, default 1)
      NUMQUANTILES .            (1-100, default 100)
  REPORT . . . . . .            (Yes/No)
  UPDATE . . . . . .            (A - All, P - Accesspath, S - Space, N - None)
  INVALIDATECACHE  .            (Yes/No)
  HISTORY  . . . . .            (A - All, P - Accesspath, S - Space, N - None)
  FORCEROLLUP  . . .            (Yes/No)
FLASHCOPY  . . . . .            (Y - Yes, N - No, C - Consistent)
KEEPDICTIONARY . . .            (Yes/No)
REUSE  . . . . . . .            (Yes/No)
LOG  . . . . . . . .            (Yes/No/NOC - NOCopypend)
WORKDDN1 . . . . . .            (DD name for temporary work file 1)
WORKDDN2 . . . . . .            (DD name for temporary work file 2)
SORTKEYS . . . . . .            (Estimated no. of keys or Yes/No)
FLOAT  . . . . . . .            (S - S390, I - IEEE)
NOSUBS . . . . . . .            (Yes/No)
ENFORCE  . . . . . .            (Yes/No)
NOCHECKPEND  . . . .            (Yes/No)
ERRDDN . . . . . . .            (DD name for error processing)
DISCARDDN  . . . . .            (DD name for discarded records)
DISCARDS . . . . . . 2          (0 to 2147483647)
SORTDEVT . . . . . .            (Device type for sort work files)
SORTNUM  . . . . . .            (Number of sort work files)
SORTWK . . . . . . .            (0-4)
IGNORE . . . . . . .            (WHEN, PART, CONV, VALPROC, IDERROR, DUPKEY)
OVERRIDE
  SYSTEMPERIOD . . .            (Yes/No)
  IDENTITY . . . . .            (Yes/No)
  TRANSID  . . . . .            (Yes/No)
  NONDETERMINISTIC .            (Yes/No)
  ROWCHANGE  . . . .            (Yes/No)
DRAIN_WAIT . . . . .            (0-1800)
RETRY  . . . . . . .            (0-255)
RETRY_DELAY  . . . .            (1-1800)
SWITCHTIME . . . . .                     (NONE, timestamp, CD, CT)
  WITH TIME ZONE . .            (Yes/No)
  YEARS  . . . . . .            (-99 - +99)
  MONTHS . . . . . .            (-99 - +99)
  DAYS . . . . . . .            (-99 - +99)
  HOURS  . . . . . .            (-99 - +99)
  MINUTES  . . . . .            (-99 - +99)
  SECONDS  . . . . .            (-99 - +99)
  MICROSECONDS . . .            (-999999 - +999999)
INDEXDEFER . . . . .            (A - ALL, N - NPI, NO - NONE)
  NONUNIQUE  . . . .            (Yes/No)
RBALRSN_CONVERSION .            (N - None, E - Extended)
DECFLOAT ROUNDING  .            (Ceiling, Down, Floor, HalfDown, HalfEven,
                                 HalfUp, Up)
IMPLICIT_TZ  . . . .            (+/-hh:mm)
```

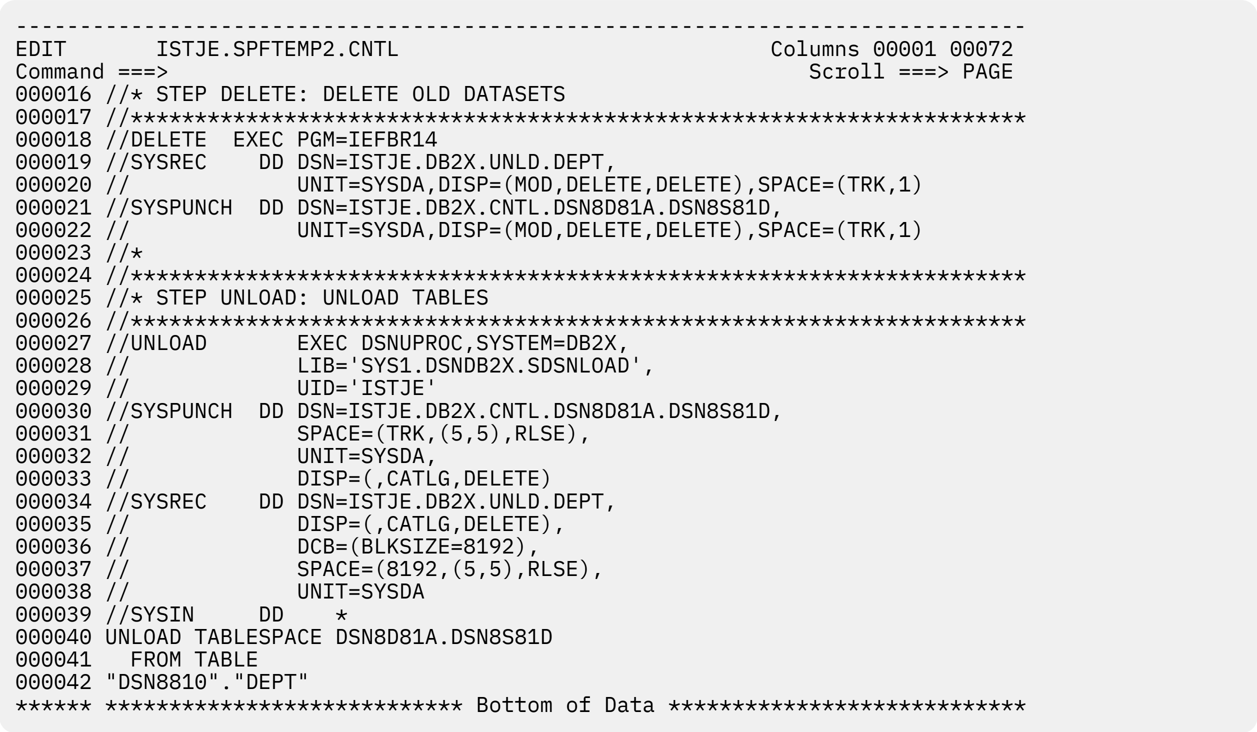
6. **If you requested a work statement list (WSL):** On the **Specify Work Statement List (ADB2WLDA)** panel, specify the WSL information, and press Enter.

   The utility statements are added to the WSL.

7. **If you requested a batch job stream:** On the ISPF edit session, change the generated job as needed or copy it to another it to another data set.

   You can use standard ISPF editor commands to manually modify the JCL.

   The following figure shows an example of the generated JCL. In this example, the **UX** option (UNLOAD using REORG UNLOAD EXTERNAL) was chosen on the **Tables Utilities (ADB2UT)** panel.



```
-------------------------------------------------------------------------------
EDIT       ISTJE.SPFTEMP2.CNTL                             Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
000016 //* STEP DELETE: DELETE OLD DATASETS
000017 //**********************************************************************
000018 //DELETE  EXEC PGM=IEFBR14
000019 //SYSREC    DD DSN=ISTJE.DB2X.UNLD.DEPT,
000020 //             UNIT=SYSDA,DISP=(MOD,DELETE,DELETE),SPACE=(TRK,1)
000021 //SYSPUNCH  DD DSN=ISTJE.DB2X.CNTL.DSN8D81A.DSN8S81D,
000022 //             UNIT=SYSDA,DISP=(MOD,DELETE,DELETE),SPACE=(TRK,1)
000023 //*
000024 //**********************************************************************
000025 //* STEP UNLOAD: UNLOAD TABLES
000026 //**********************************************************************
000027 //UNLOAD       EXEC DSNUPROC,SYSTEM=DB2X,
000028 //             LIB='SYS1.DSNDB2X.SDSNLOAD',
000029 //             UID='ISTJE'
000030 //SYSPUNCH  DD DSN=ISTJE.DB2X.CNTL.DSN8D81A.DSN8S81D,
000031 //             SPACE=(TRK,(5,5),RLSE),
000032 //             UNIT=SYSDA,
000033 //             DISP=(,CATLG,DELETE)
000034 //SYSREC    DD DSN=ISTJE.DB2X.UNLD.DEPT,
000035 //             DISP=(,CATLG,DELETE),
000036 //             DCB=(BLKSIZE=8192),
000037 //             SPACE=(8192,(5,5),RLSE),
000038 //             UNIT=SYSDA
000039 //SYSIN     DD    *
000040 UNLOAD TABLESPACE DSN8D81A.DSN8S81D
000041   FROM TABLE
000042 "DSN8810"."DEPT"
****** **************************** Bottom of Data ****************************
```

Figure 331. Edit generated JCL panel—UNLOAD utility (ADB2UE)

8. Make the following changes to the generated utility statements as needed:

   - If the utility statement unloads a table and might produce a record length that exceeds 32K, change the utility options so that the record length is less than 32K. (The record length might exceed 32K if the table has LOB objects.) Db2 Admin Tool requires that the record length be less than 32K when unloading tables.
   - If you specified **UX** and **Generate work statement list** = YES, add a TEMPLATE utility statement to the WSL. The generated REORG statement references a DD name but does not include a template for it.

9. Submit the utility job or run the WSL.

Related tasks

"Running a WSL" on page 551
You can run a work statement list (WSL) either in batch or online.

Running Db2 utilities on indexes

Some Db2 utilities can be run against indexes.

About this task

If you want to run a utility against a LISTDEF list of indexes, see "Running utilities on LISTDEF lists" on page 609.

Procedure

To run Db2 utilities on indexes:

1. Select the indexes on which you want to run the utility:

   a) On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.

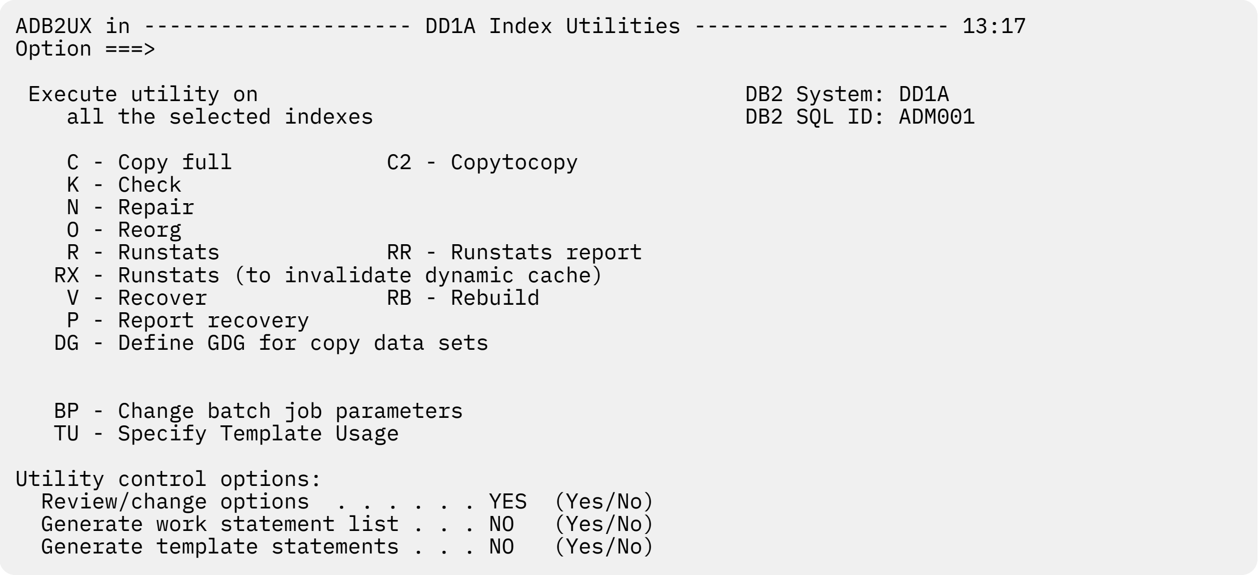
   b) On the **System Catalog (ADB21)** panel, if you want to select all indexes in one or more databases, select option D. Otherwise, select option X. Optionally specify any filtering criteria at the bottom of the panel, and press Enter.

   c) If the **Databases (ADB21D)** panel is displayed, filter the list as needed, and specify the UTIL IX primary command.

   d) On the **Indexes (ADB21X)** panel, select one or more indexes:

      - If you want to select one index, issue the UT line command next to the table name, and press Enter.
      - If you want to select more than one index, filter the list as needed, specify the UTIL primary command, and press Enter.

   The **Index Utilities (ADB2UX)** panel is displayed:



```
ADB2UX in --------------------- DD1A Index Utilities -------------------- 13:17
Option ===>

 Execute utility on                                      DB2 System: DD1A
    all the selected indexes                             DB2 SQL ID: ADM001

    C - Copy full            C2 - Copytocopy
    K - Check
    N - Repair
    O - Reorg
    R - Runstats             RR - Runstats report
   RX - Runstats (to invalidate dynamic cache)
    V - Recover              RB - Rebuild
    P - Report recovery
   DG - Define GDG for copy data sets


   BP - Change batch job parameters
   TU - Specify Template Usage

Utility control options:
  Review/change options  . . . . . . YES  (Yes/No)
  Generate work statement list . . . NO   (Yes/No)
  Generate template statements . . . NO   (Yes/No)
```

*Figure 332. **Index Utilities (ADB2UX)** panel*

2. Optional: Specify the following options as needed:

   BP

   If you plan to generate a batch utility job stream and want to change the JOB statement or other system parameters, specify **BP** and press Enter. On the resulting **Batch Job Utility Parameters (ADB2UPA)** panel, specify the options that you want to change. Then, press PF3 to save your changes and return to the **Tables Utilities (ADB2UT)** panel.

   TU

   If you plan to use templates and want to specify which template to use for a given utility option, specify **TU**. Then, follow the instructions in “Associating templates with data sets” on page 439.

   Tip: When you run the COPY utility, by default one copy is written to the data set that is defined in the SYSCOPY DD statement. If you want more than one copy of the output, you can create and use templates for COPYDDN 1, COPYDDN2, RECOVERYDDN1, and RECOVERYDDN2.

3. In the **Option** field, specify a utility to run on the selected index. For example, specify R to run the RUNSTATS utility.

4. Change any of the following control options, and press Enter:

Review/change options

Specify whether you want to review and change the utility options. If the value is NO, the default options for the selected utility are used.

For the REPAIR utility (option N), you must select which REPAIR function you want to run. Therefore, Db2 Admin Tool always uses a value of YES for this field and displays a subsequent option panel, **Specify Utility Options - REPAIR INDEX (ADB2UXN)** panel, even if you set this field to NO.

Generate work statement list

Specify whether you want the utility control statements to be added to a work statement list (WSL). If the value is NO, an executable utility job stream is generated instead.

When you specify the CHECK utility, a batch statement list, which is similar to a WSL, is generated by default, regardless of the value of the **Generate work statement list** field. The batch statement list is required as an input file to the Batch Restart program (ADBTEP2), which manages the CHECK utility function.

Generate template statements

Specify whether you want templates to be used. If the value is YES, Db2 Admin Tool uses the active templates that are defined. (If you want to change these templates, specify the **TU** option.)

Db2 Admin Tool does not generate any TSODELETE statements to delete any existing data sets for the template first. To ensure that any existing data sets are deleted, consider using one of the following techniques when you define the template:

- Specify the data set name pattern as a GDG (generation data group) where the next data set in the sequence is generated (+1), and change the other common options so that the GDGLIMIT is 1. This setup causes the data sets in the group to roll off so that only one data set exists at any one time. For example, a data set pattern name might be specified as &db..&ts..&name..ic(+1).
- Specify a DISP option of NEW, DELETE, DELETE for the data set, if appropriate.

When you specify the CHECK utility, templates are used regardless of the value of the **Generate template statements** field, because the CHECK utility function requires the use of templates. Either the default templates or the templates that you specify are used.

5. If the **Specify Utility Options** panel is displayed, specify any additional options, and press Enter.

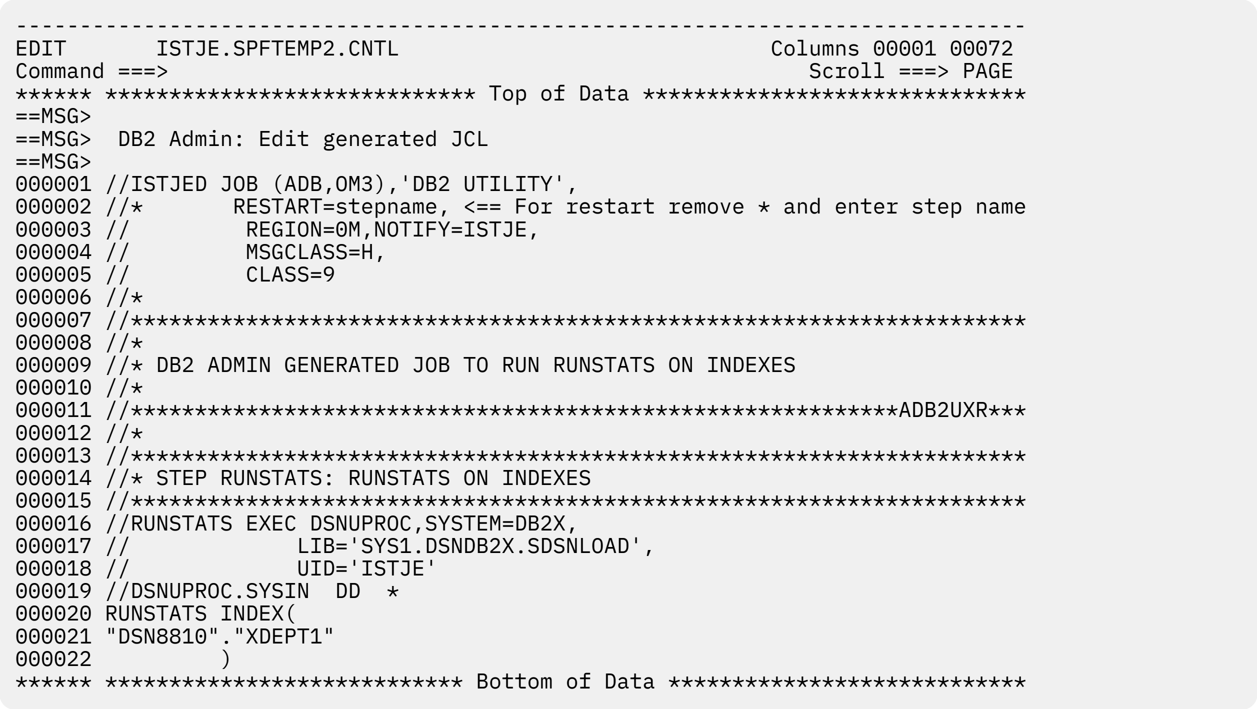
   This panel is displayed if **Review/change options** = YES or if the utility selected is N (REPAIR).

   You must specify the options according to the Db2 syntax rules for utilities. For additional information, restrictions, and recommendations on various utility options, see Db2 online utilities (Db2 12 for z/OS).
6. **If you requested a work statement list (WSL):** On the **Specify Work Statement List (ADB2WLDA)** panel, specify the WSL information, and press Enter.

   The utility statements are added to the WSL.
7. **If you requested a batch job stream:** On the ISPF edit session, change the generated job as needed or copy it to another it to another data set.

   You can use standard ISPF editor commands to manually modify the JCL.

   The following figure shows an example of the generated JCL. In this example, the **R** option (RUNSTATS) was chosen on the **Index Utilities (ADB2UX)** panel.



```
-------------------------------------------------------------------------------
EDIT       ISTJE.SPFTEMP2.CNTL                            Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
****** ***************************** Top of Data ******************************
==MSG>
==MSG>  DB2 Admin: Edit generated JCL
==MSG>
000001 //ISTJED JOB (ADB,OM3),'DB2 UTILITY',
000002 //*       RESTART=stepname, <== For restart remove * and enter step name
000003 //         REGION=0M,NOTIFY=ISTJE,
000004 //         MSGCLASS=H,
000005 //         CLASS=9
000006 //*
000007 //**********************************************************************
000008 //*
000009 //* DB2 ADMIN GENERATED JOB TO RUN RUNSTATS ON INDEXES
000010 //*
000011 //************************************************************ADB2UXR***
000012 //*
000013 //**********************************************************************
000014 //* STEP RUNSTATS: RUNSTATS ON INDEXES
000015 //**********************************************************************
000016 //RUNSTATS EXEC DSNUPROC,SYSTEM=DB2X,
000017 //             LIB='SYS1.DSNDB2X.SDSNLOAD',
000018 //             UID='ISTJE'
000019 //DSNUPROC.SYSIN  DD  *
000020 RUNSTATS INDEX(
000021 "DSN8810"."XDEPT1"
000022          )
****** **************************** Bottom of Data ****************************
```

Figure 333. Edit generated JCL panel—RUNSTATS utility (ADB2UE)

8. Submit the utility job or run the WSL.

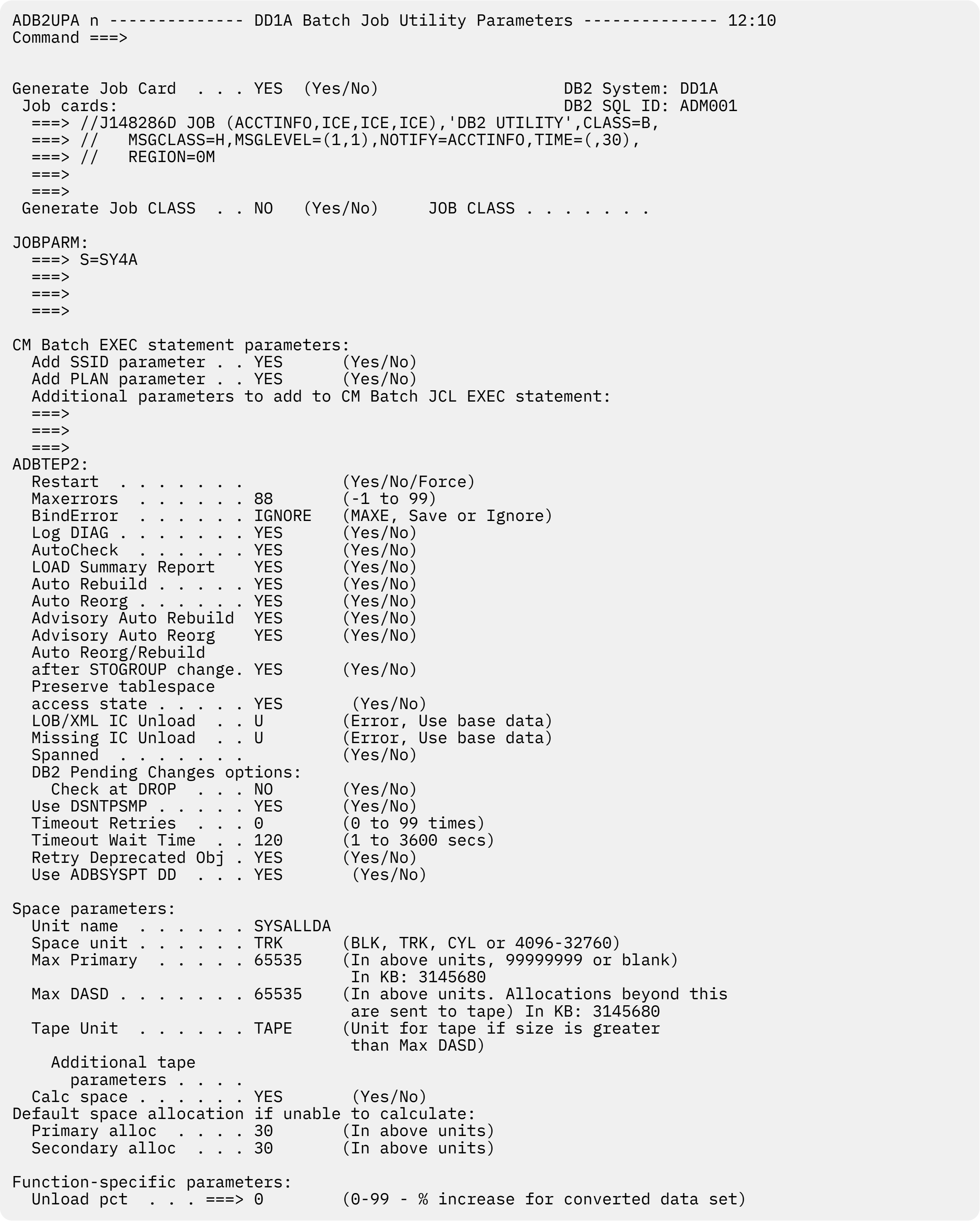
Related tasks

"Running a WSL" on page 551
You can run a work statement list (WSL) either in batch or online.

Batch job parameters for utility jobs

When you run utilities, you can change some of the batch job parameters, such as the JOB card, the EXEC statement parameters, the ADBTEP2 parameters, and the space parameters.

Changing these parameters can be done when running utilities on tables spaces, tables, or indexes. When you specify **BP** on the **Table Space Utilities (ADB2US)** panel, the **Tables Utilities (ADB2UT)** panel, or the **Index Utilities (ADB2UX)** panel, the **Batch Job Utility Parameters (ADB2UPA)** panel is displayed:



```
ADB2UPA n ------------- DD1A Batch Job Utility Parameters ------------- 12:10
Command ===>


Generate Job Card  . . . YES  (Yes/No)                  DB2 System: DD1A
 Job cards:                                             DB2 SQL ID: ADM001
  ===> //J148286D JOB (ACCTINFO,ICE,ICE,ICE),'DB2 UTILITY',CLASS=B,
  ===> //   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=ACCTINFO,TIME=(,30),
  ===> //   REGION=0M
  ===>
  ===>
 Generate Job CLASS  . . NO   (Yes/No)     JOB CLASS . . . . . . .

JOBPARM:
  ===> S=SY4A
  ===>
  ===>
  ===>

CM Batch EXEC statement parameters:
  Add SSID parameter . . YES      (Yes/No)
  Add PLAN parameter . . YES      (Yes/No)
  Additional parameters to add to CM Batch JCL EXEC statement:
  ===>
  ===>
  ===>
ADBTEP2:
  Restart  . . . . . . .          (Yes/No/Force)
  Maxerrors  . . . . . . 88       (-1 to 99)
  BindError  . . . . . . IGNORE   (MAXE, Save or Ignore)
  Log DIAG . . . . . . . YES      (Yes/No)
  AutoCheck  . . . . . . YES      (Yes/No)
  LOAD Summary Report    YES      (Yes/No)
  Auto Rebuild . . . . . YES      (Yes/No)
  Auto Reorg . . . . . . YES      (Yes/No)
  Advisory Auto Rebuild  YES      (Yes/No)
  Advisory Auto Reorg    YES      (Yes/No)
  Auto Reorg/Rebuild
  after STOGROUP change. YES      (Yes/No)
  Preserve tablespace
  access state . . . . . 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)
  Use DSNTPSMP . . . . . YES      (Yes/No)
  Timeout Retries  . . . 0        (0 to 99 times)
  Timeout Wait Time  . . 120      (1 to 3600 secs)
  Retry Deprecated Obj . YES      (Yes/No)
  Use ADBSYSPT DD  . . . YES       (Yes/No)

Space parameters:
  Unit name  . . . . . . SYSALLDA
  Space unit . . . . . . TRK      (BLK, TRK, CYL or 4096-32760)
  Max Primary  . . . . . 65535    (In above units, 99999999 or blank)
                                   In KB: 3145680
  Max DASD . . . . . . . 65535    (In above units. Allocations beyond this
                                   are sent to tape) In KB: 3145680
  Tape Unit  . . . . . . TAPE     (Unit for tape if size is greater
                                   than Max DASD)
    Additional tape
      parameters . . . .
  Calc space . . . . . . YES       (Yes/No)
Default space allocation if unable to calculate:
  Primary alloc  . . . . 30       (In above units)
  Secondary alloc  . . . 30       (In above units)

Function-specific parameters:
  Unload pct  . . . ===> 0        (0-99 - % increase for converted data set)
```

*Figure 334. **Batch Job Utility Parameters (ADB2UPA)** panel*

The **Batch Job Utility Parameters (ADB2UPA)** panel has the following options:

Generate Job Card

Specify whether you want to generate the JOB card. If you choose to generate a JOB card, you can also generate the CLASS parameter. If you specify a JOB CLASS, the last line of the JOB card must end with a comma, because Db2 Admin Tool adds an additional line to the JOB card for the JOB CLASS.

Generate Job CLASS

Specify whether you want to generate the CLASS parameter. If you specify YES, you can specify a JOB CLASS to override the JOB CLASS that is specified by the installation.

JOBPARM

Specify a JOBPARM statement. If JOBPARM is not specified on this panel, Db2 Admin Tool adds a line for the installation-specified JOBPARM.

CM Batch EXEC statement parameters

Customize the following JCL parameters for invoking CM batch:

Add SSID parameter

Specify whether to add the SSID parameter to the EXEC statement. YES is the default value.

Add PLAN parameter

Specify whether to add the PLAN parameter to the EXEC statement. YES is the default value.

Additional parameters to add to CM Batch JCL EXEC statement

Specify additional parameters to add to the EXEC statement. The CM Batch JCL procedure must be predefined to accept any additional JCL procedure parameters.

Use the syntax: *parameter_name=value*, where *parameter_name* is the name of the parameter and *value* is its value.

The following examples illustrate how you might customize the CM batch JCL parameters and the resulting JCL EXEC statements.

Example 1: Suppose that you specify the following parameters on the **Batch Job Utility Parameters (ADB2UPA)** panel:

```
Add SSID parameter . . YES (Yes,No)
Add PLAN parameter . . YES (Yes,No)
Additional parameters to add to CM Batch JCL EXEC statement:
    ===>
    ===>
    ===>
```

The following JCL EXEC statement is generated:

```
//GOCCM    EXEC GOCCM,
// SSID=DSNA,
// PLAN=ADB
```

Example 2: Suppose that you specify the following parameters on the **Batch Job Utility Parameters (ADB2UPA)** panel:

```
Add SSID parameter . . NO (Yes,No)
Add PLAN parameter . . NO (Yes,No)
Additional parameters to add to CM Batch JCL EXEC statement:
    ===>
    ===>
    ===>
```

The following JCL EXEC statement is generated:

```
JCL EXEC statement used to invoke CM Batch:
//GOCCM    EXEC GOCCM
```

Example 3: Suppose that you specify the following parameters on the **Batch Job Utility Parameters (ADB2UPA)** panel:

```
Add SSID parameter . . NO  (Yes,No)
Add PLAN parameter . . NO  (Yes,No)
Additional parameters to add to CM Batch JCL EXEC statement:
    ===> PROFILE=DSNA
    ===> PROFILE2=ABC
    ===>
```

The following JCL EXEC statement is generated:

```
JCL EXEC statement used to invoke CM Batch:
//GOCCM    EXEC GOCCM
```

ADBTEP2:

Customize options for the batch restart program, ADBTEP2.

The fields listed under **ADBTEP2:** on this panel set values for various ADBTEP2 parameters. The exception is the **Use ADBSYSPT DD** field, which controls the JCL that is used to call ADBTEP2. For detailed descriptions of the ADBTEP2 fields and the corresponding parameters other than **Use ADBSYSPT DD**, see "Parameters passed to the ADBTEP2 program" on page 573.

Use ADBSYSPT DD

Specifies whether an ADBSYSPT DD statement is added to the ADBTEP2 step in the JCL.

An ADBSYSPT DD allows for a separate SYSOUT file for ADBTEP2 output. Otherwise, ADBTEP2 output is written to the SYSPRINT DD, which is also used by Db2 utilities for its output. Separating this output can help readability and prevent potential errors due to more than one application using SYSPRINT DD for output but with different file attributes.

YES

ADBSYSPT is added to the JCL, as follows:

```
//ADBSYSPT DD SYSOUT=*
```

NO

ADBSYSPT is not added to the JCL.

For systems that use JES3, the default is YES and this field is not displayed. Otherwise, the default is NO.

Space parameters:

Unit name

Specify the default unit name to use for allocating new data sets.

Space unit

Specify the unit in which space is to be allocated. You can specify that space be allocated in any of the following units:

BLK

Blocks

TRK

Tracks

CYL

Cylinders

integer

A specified number of kilobytes in the range of 4096 to 32760.

Max Primary

Specify the maximum amount of primary space that can be allocated for a data set on DASD, as measured in the unit that is specified for **Space unit**.

Max DASD
Specify the maximum amount of space that can be allocated for a data set on DASD, as measured in the unit that is specified for **Space unit**. When Db2 Admin Tool estimates that the amount of space that is required for a data set exceeds this value, the data set is allocated to tape.

Tape unit
Specify a valid tape unit that is defined at your site.

Additional tape parameters
Specify other parameters for allocating utility data sets on tape.

Calc space
Specify whether to calculate space for utilities.

Primary alloc
Specify the default size for primary space allocation when Db2 Admin Tool cannot estimate the space requirements for an allocated data set. This situation can occur when the RUNSTATS and STOSPACE utilities have not been run.

Secondary alloc
Specify the default size for secondary space allocation when Db2 Admin Tool cannot estimate the space requirements for an allocated data set. This situation can occur when the RUNSTATS and STOSPACE utilities have not been run.

Unload pct
Specify the percentage increase in data set size for the UNLOAD data set.

The ALT function and Object Compare function convert data from the UNLOAD step. This converted data might require more space than the UNLOAD data set. **Unload pct** lets you increase the size of the converted data set by a percentage of the UNLOAD data set. This action can avoid out-of-space conditions.

Related tasks

"Running Db2 utilities on table spaces " on page 594
Many Db2 utilities run against table spaces. You can use Db2 Admin Tool to specify the utility options and generate the JCL to run these utilities.

"Running Db2 utilities on tables" on page 598
You can use Db2 Admin Tool to run the LOAD, UNLOAD, and REORG UNLOAD EXTERNAL utilities on tables. Based on the options that you choose, Db2 Admin Tool generates utility statements for you. You can choose to either save these utility statements in a work statement list (WSL) or generate a batch utility job stream.

"Running Db2 utilities on indexes " on page 602
Some Db2 utilities can be run against indexes.

Running utilities on LISTDEF lists

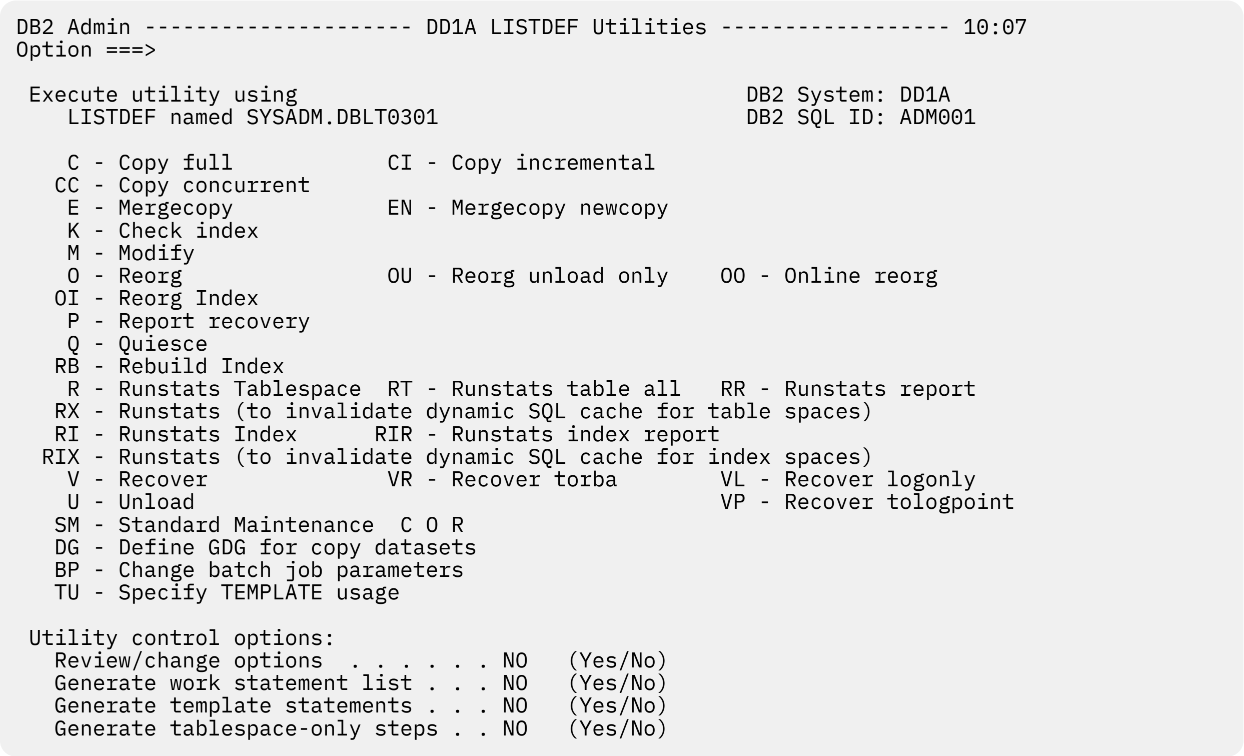
Instead of running utilities against explicitly specified table spaces or indexes, you can run utilities against a predefined list of objects. These lists are defined by the LISTDEF utility.

Procedure

To run utilities on a predefined LISTDEF list:

1. On the **DB2 Administration Menu (ADB2)** panel, select option 5.
2. On the **Utility generation using LISTDEFs and TEMPLATEs (ADB25)** panel, specify L.
3. On the **LISTDEFs (ADB25L)** panel, specify the UT line command next to the LISTDEF list that contains the objects on which you want to run a utility.

   The **LISTDEF Utilities (ADB25LU)** panel is displayed:



```
DB2 Admin --------------------- DD1A LISTDEF Utilities ------------------ 10:07
Option ===>

 Execute utility using                                   DB2 System: DD1A
    LISTDEF named SYSADM.DBLT0301                        DB2 SQL ID: ADM001

    C - Copy full            CI - Copy incremental
   CC - Copy concurrent
    E - Mergecopy            EN - Mergecopy newcopy
    K - Check index
    M - Modify
    O - Reorg                OU - Reorg unload only    OO - Online reorg
   OI - Reorg Index
    P - Report recovery
    Q - Quiesce
   RB - Rebuild Index
    R - Runstats Tablespace  RT - Runstats table all   RR - Runstats report
   RX - Runstats (to invalidate dynamic SQL cache for table spaces)
   RI - Runstats Index      RIR - Runstats index report
  RIX - Runstats (to invalidate dynamic SQL cache for index spaces)
    V - Recover              VR - Recover torba        VL - Recover logonly
    U - Unload                                         VP - Recover tologpoint
   SM - Standard Maintenance  C O R
   DG - Define GDG for copy datasets
   BP - Change batch job parameters
   TU - Specify TEMPLATE usage

 Utility control options:
   Review/change options  . . . . . . NO   (Yes/No)
   Generate work statement list . . . NO   (Yes/No)
   Generate template statements . . . NO   (Yes/No)
   Generate tablespace-only steps . . NO   (Yes/No)
```

*Figure 335. **LISTDEF Utilities (ADB25LU)** panel*

4. Optional: Specify the following options as needed:

   BP

   If you plan to generate a batch utility job stream and want to change the JOB statement or other system parameters, specify **BP** and press Enter. On the resulting **Batch Job Utility Parameters (ADB2UPA)** panel, specify the options that you want to change. Then, press PF3 to save your changes and return to the **Tables Utilities (ADB2UT)** panel.

   TU

   If you plan to use templates and want to specify which template to use for a given utility option, specify **TU**. Then, follow the instructions in "Associating templates with data sets" on page 439.

5. In the **Option** field, specify a utility to run on the selected table. For example, specify C to run the COPY utility.
6. Change any of the following control options, and press Enter:

   Review/change options

   Specify whether you want to review and change the utility options. If the value is NO, the default options for the selected utility are used.

   Generate work statement list

   Specify whether you want the utility control statements to be added to a work statement list (WSL). If the value is NO, an executable utility job stream is generated instead.

   Generate template statements

   Specify whether you want templates to be used. If the value is YES, Db2 Admin Tool uses the active templates that are defined. (If you want to change these templates, specify the **TU** option.)

   Db2 Admin Tool does not generate any TSODELETE statements to delete any existing data sets for the template first. To ensure that any existing data sets are deleted, consider using one of the following techniques when you define the template:

   - Specify the data set name pattern as a GDG (generation data group) where the next data set in the sequence is generated (+1), and change the other common options so that the GDGLIMIT is 1. This setup causes the data sets in the group to roll off so that only one data set exists at any one time. For example, a data set pattern name might be specified as &db..&ts..&name..ic(+1).

- Specify a DISP option of NEW, DELETE, DELETE for the data set, if appropriate.

7. If the **Specify Utility Options** panel is displayed, specify any additional options, and press Enter.

   This panel is displayed if **Review/change options** = YES.

   You must specify the options according to the Db2 syntax rules for utilities. For additional information, restrictions, and recommendations on various utility options, see Db2 online utilities (Db2 12 for z/OS).
8. **If you requested a work statement list (WSL):** On the **Specify Work Statement List (ADB2WLDA)** panel, specify the WSL information, and press Enter.

   The utility statements are added to the WSL.
9. **If you requested a batch job stream:** On the ISPF edit session, change the generated job as needed or copy it to another it to another data set.

   You can use standard ISPF editor commands to manually modify the JCL.
10. Review and submit the generated JCL or run the specified WSL to run the utility.

Related concepts

"LISTDEF and TEMPLATE" on page 416
LISTDEF and TEMPLATE are Db2 utilities that provide facilities for other utilities. You can use LISTDEF to define reusable lists of objects for other utilities to process. You can use TEMPLATE to define templates for data sets that are allocated by other utilities. LISTDEF and TEMPLATE are often used together.

Related tasks

"Running a WSL" on page 551
You can run a work statement list (WSL) either in batch or online.

Utility options for XML and LOBs

Some Db2 utility options support XML and LOBs.

The following Db2 utility options support XML and LOBs:

CHECK DATA

- Option XMLERROR can provide the values REPORT and INVALIDATE on XML column checks.
- Option PUNCH DD is applicable only when SHRLEVEL is specified as CHANGE. For XML table spaces, before running CHECK DATA, PUNCHDD runs CHECK INDEX on the node ID index of each XML column.
- Option LOBERROR provides the values REPORT and INVALIDATE on LOB column checks.
- Option CLONE indicates that CHECK DATA is to check the clone table in the specified table space. Because clone tables cannot have referential constraints, the utility checks only constraints for inconsistencies between the clone table data and the corresponding LOB data. If you do not specify CLONE, CHECK DATA operates only against the base table.

CHECK INDEX
OPTION CLONE

COPY
OPTION CLONE

COPYTOCOPY
OPTION CLONE

LISTDEF
LOB and XML types are supported.

REBUILD INDEX
REBUILD INDEX with SHRLEVEL CHANGE is not allowed for XML Indexes.

REORG
For XML table spaces, and base tables with XML columns, you cannot specify the following options in a REORG statement: DISCARD, REBALANCE, and UNLOAD EXTERNAL.

Running Db2 stand-alone utilities

You can run the following Db2 stand-alone utilities in Db2 Admin Tool: DSN1COMP, DSN1COPY, and DSN1PRNT. Db2 *stand-alone utilities* run independently of Db2; they work directly on the data sets. In Db2 Admin Tool, these utilities are also known as *offline utilities*.

Procedure

To run Db2 stand-alone utilities:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. From the **System Catalog (ADB21)** panel, navigate to the **Offline Utilities Selection (ADB2US1)** panel by completing one of the following procedures:

   Procedure 1: For table space data sets:

   a. Specify option S, and press Enter.
   b. On the **Table Spaces (ADB21S)** panel, specify the SP line command, and press Enter.
   c. On the **Table Space Parts (ADB21SP)** panel, specify the UT line command, and press Enter.
   d. On the **Table Space Utilities (ADB2US)** panel, specify option DSN1, and press Enter.

   Procedure 2: For index space data sets:

   a. Specify option X, and press Enter.
   b. On the **Indexes (ADB21X)** panel, specify the XP line command, and press Enter.
   c. On the **Index Parts (ADB21XP)** panel, specify the UT line command, and press Enter.
   d. On the **Index Utilities (ADB2UX)** panel, specify option DSN1, and press Enter.
3. On the **Offline Utilities Selection (ADB2US1)** panel, specify one of the following offline utilities to run, and press Enter:

   1P
   DSN1PRNT – Print the following types of data sets:

   - Db2 VSAM data sets that contain table spaces or index spaces
   - Image copy data sets
   - Sequential data sets that contain Db2 table spaces or index spaces

   1C
   DSN1COPY – Copy the following types of data sets:

   - Copy Db2 VSAM data sets to sequential data sets
   - Copy DSN1COPY sequential data sets to Db2 VSAM data sets
   - Copy Db2 image copy data sets to Db2 VSAM data sets
   - Copy Db2 VSAM data sets to other Db2 VSAM data sets
   - Copy DSN1COPY sequential data sets to other sequential data sets

   The 1C option requires an output data set, defined by a SYSUT2 DD statement. If you do not specify an output data set, Db2 Admin Tool defaults to DUMMY. If you specify an existing data set (DISP=OLD), provide the name and disposition. For a new data set (DISP=NEW), you must also specify, at a minimum, the space units (either TRK or CYL). You can also provide the primary and secondary space allocations and the unit type.

   1M
   DSN1COMP – Estimate space savings as a result of Db2 data compression in table spaces. This option is not available for index spaces.
4. On the **Offline Utilities Parameters (ADB2USOF)** panel, specify the appropriate parameter values and press Enter.
5. Review and edit the generated JCL in the ISPF edit session.

6. Submit the job.

Related information

Db2 stand-alone utilities (Db2 12 for z/OS)
DSN1PRNT (Db2 12 for z/OS)
DSN1COPY (Db2 12 for z/OS)
DSN1COMP (Db2 12 for z/OS)

Unloading objects by using Db2 High Performance Unload

You can use Db2 High Performance Unload with the MIG function to unload Db2 objects more efficiently.

Before you begin

You must have Db2 High Performance Unload installed.

About this task

Although Db2 Admin Tool does not run HPU in z/OS storage key 7, this situation does not cause any problems in running HPU. If you receive warning message INZU241I from HPU, you can ignore this message unless abends occur.

Procedure

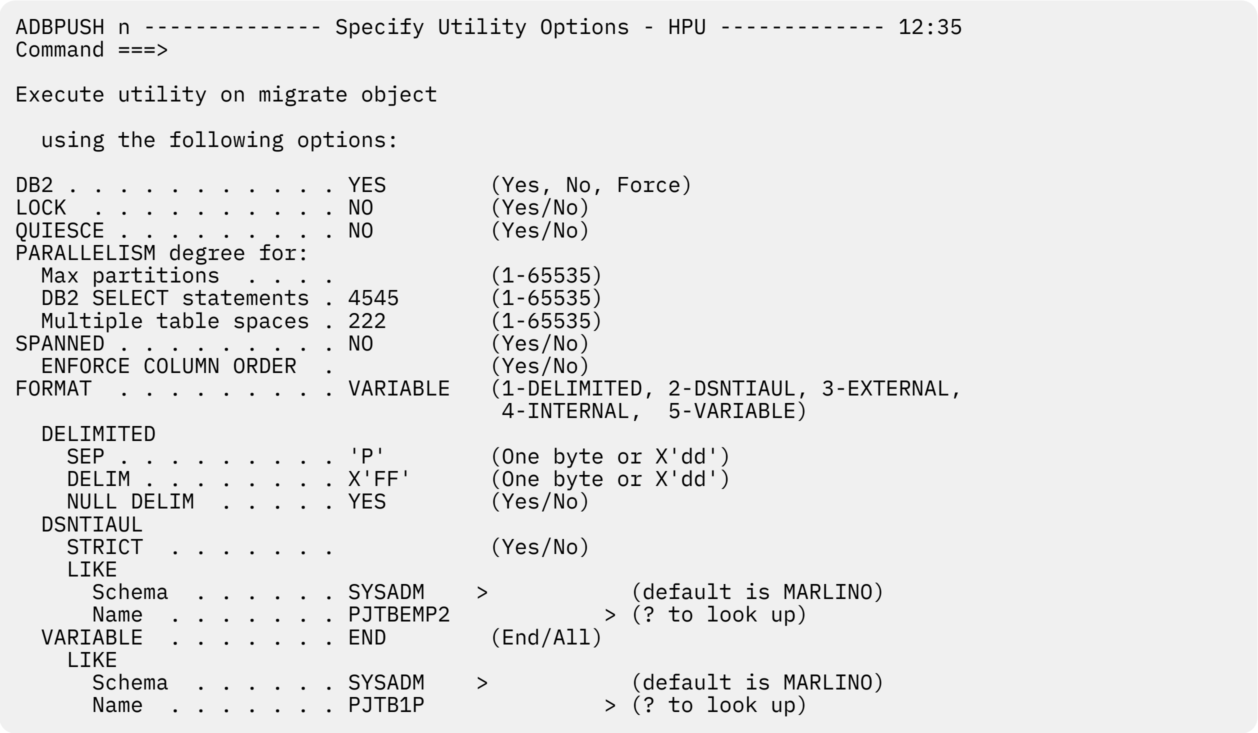
To unload objects by using Db2 High Performance Unload :

1. On the **Migrate Table Spaces (ADB28S)** panel, issue the NEXT primary command.
2. On the **Migrate Parameters (ADB28M)** panel, issue the UO primary command.
3. On the **Change Utilities Options (ADB2UOPS)** panel, enter the HPU option.

```
ADB2UOPS  ---------------- Change Utilities Options ---------------- 12:29
Option ===>

Select one of the following, then press Enter.

   C - Image copy
  KD - Check data
   M - Modify
   O - Reorg tablespace
  OI - Reorg index
   R - Runstats tablespace
   U - Unload
 HPU - High Performance Unload
   L - Load
```

Figure 336. ***Change Utilities Options (ADB2UOPS)*** *panel*

4. On the **Specify Utility Options - HPU (ADBPUSH)** panel, specify values for your HPU parameters.



```
ADBPUSH n -------------- Specify Utility Options - HPU ------------ 12:35
Command ===>

Execute utility on migrate object

  using the following options:

DB2 . . . . . . . . . . . YES        (Yes, No, Force)
LOCK  . . . . . . . . . . NO         (Yes/No)
QUIESCE . . . . . . . . . NO         (Yes/No)
PARALLELISM degree for:
  Max partitions  . . . .            (1-65535)
  DB2 SELECT statements . 4545       (1-65535)
  Multiple table spaces . 222        (1-65535)
SPANNED . . . . . . . . . NO         (Yes/No)
  ENFORCE COLUMN ORDER  .            (Yes/No)
FORMAT  . . . . . . . . . VARIABLE   (1-DELIMITED, 2-DSNTIAUL, 3-EXTERNAL,
                                      4-INTERNAL,  5-VARIABLE)
  DELIMITED
    SEP . . . . . . . . . 'P'        (One byte or X'dd')
    DELIM . . . . . . . . X'FF'      (One byte or X'dd')
    NULL DELIM  . . . . . YES        (Yes/No)
  DSNTIAUL
    STRICT  . . . . . . .            (Yes/No)
    LIKE
      Schema  . . . . . . SYSADM    >           (default is MARLINO)
      Name  . . . . . . . PJTBEMP2          > (? to look up)
  VARIABLE  . . . . . . . END        (End/All)
    LIKE
      Schema  . . . . . . SYSADM    >           (default is MARLINO)
      Name  . . . . . . . PJTB1P            > (? to look up)
```

Figure 337. ***Specify Utility Options - HPU (ADBPUSH)*** *panel*

5. Press PF3 to return to the **Migrate Parameters (ADB28M)** panel, where you can continue with the MIG process.

Db2 EXPLAIN

You can use Db2 Admin Tool to issue SQL EXPLAIN statements, which gather information about the access path that Db2 chooses to process a query.

Specifically, you can use the **Explain** option in Db2 Admin Tool to do the following tasks:

- Explain an SQL statement and display the resulting rows in a plan table. You can use this information to see how Db2 will run SQL statements in application plans or packages that were bound with EXPLAIN(YES).
- Create any needed Db2 EXPLAIN tables.
- Create an alias for the EXPLAIN tables.
- Create indexes on any EXPLAIN tables. An index on the plan table is recommended if optimizer hints are being used.
- Upgrade any EXPLAIN tables to the current version of Db2.
- List queries in the SYSQUERY table.
- Issue the Db2 BIND QUERY command on queries in SYSQUERY. The BIND QUERY panel supports the EXPLAININPUTSCHEMA() clause, which allows you to copy specified rows from an overpopulated plan table to one that should be used solely for BIND QUERY.

Explaining SQL Statements

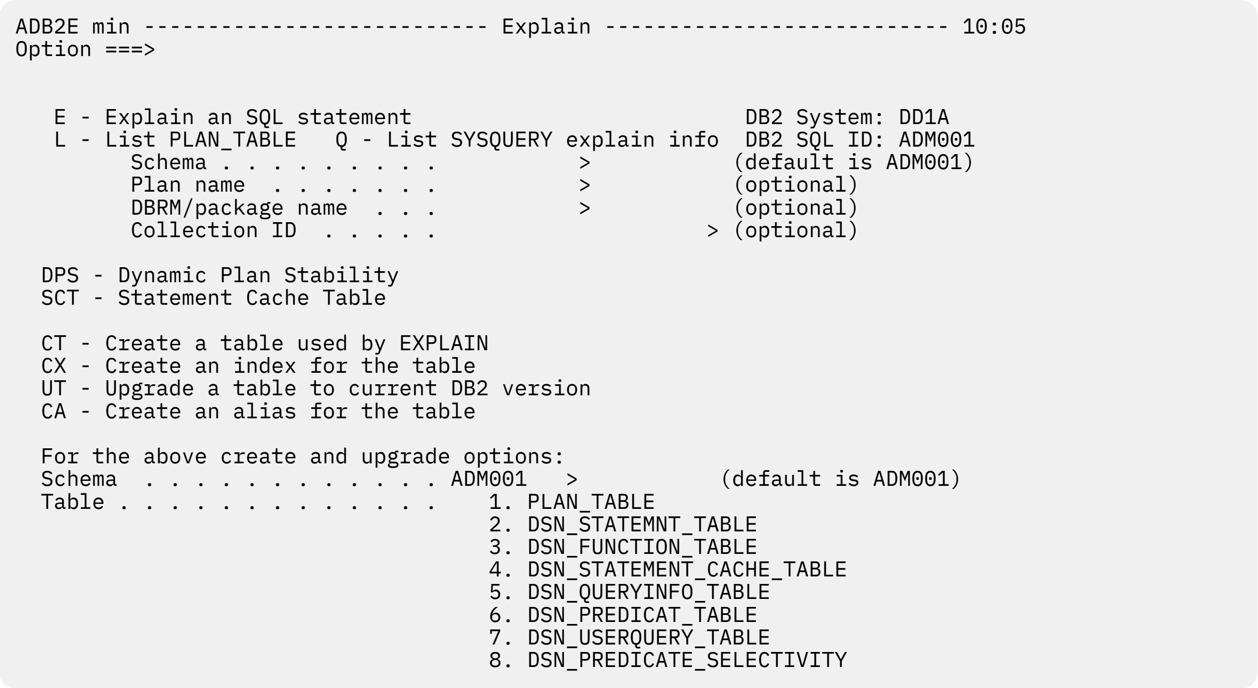
You can use Db2 EXPLAIN to capture access path information for your queries.

Procedure

To explain an SQL statement:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option E, and press Enter.

2. On the **EXPLAIN (ADB2E)** panel, specify option E, and press Enter:



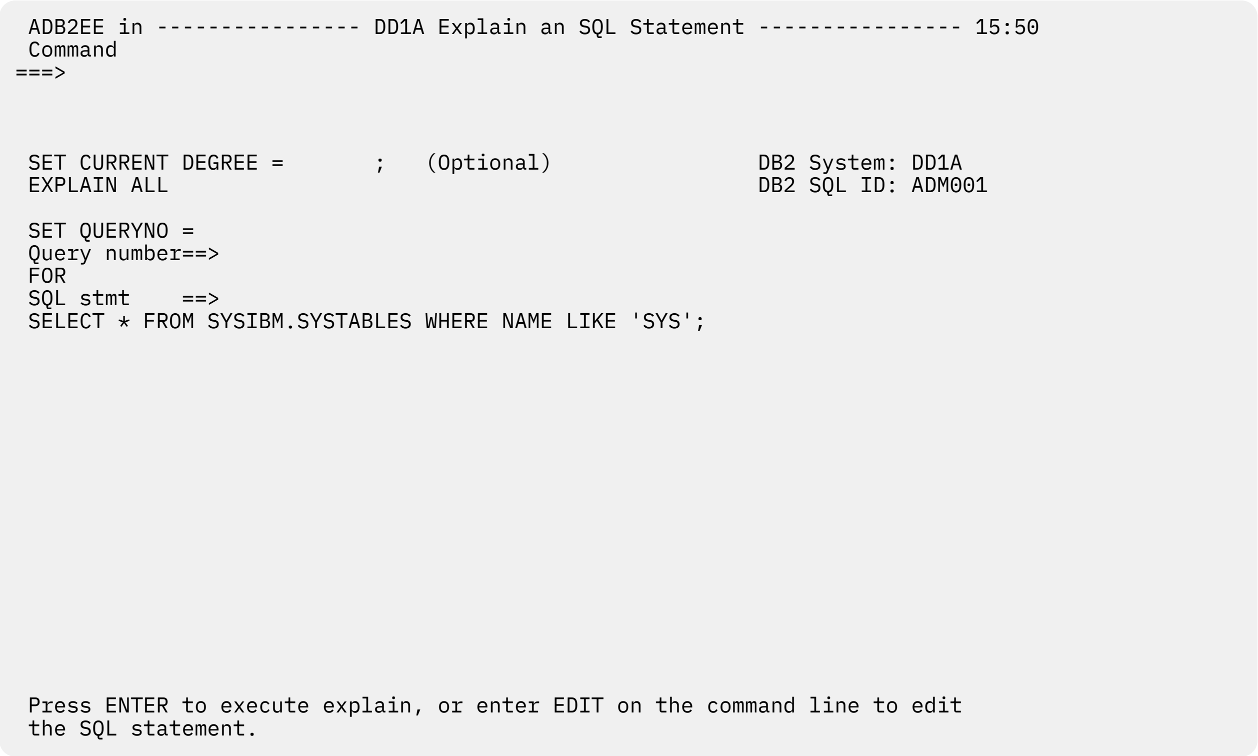
```
ADB2E min --------------------------- Explain --------------------------- 10:05
Option ===>


   E - Explain an SQL statement                          DB2 System: DD1A
   L - List PLAN_TABLE   Q - List SYSQUERY explain info  DB2 SQL ID: ADM001
         Schema . . . . . . . . .           >           (default is ADM001)
         Plan name  . . . . . . .           >           (optional)
         DBRM/package name  . . .           >           (optional)
         Collection ID  . . . . .                     > (optional)

  DPS - Dynamic Plan Stability
  SCT - Statement Cache Table

  CT - Create a table used by EXPLAIN
  CX - Create an index for the table
  UT - Upgrade a table to current DB2 version
  CA - Create an alias for the table

  For the above create and upgrade options:
  Schema  . . . . . . . . . . . . ADM001   >           (default is ADM001)
  Table . . . . . . . . . . . . .    1. PLAN_TABLE
                                     2. DSN_STATEMNT_TABLE
                                     3. DSN_FUNCTION_TABLE
                                     4. DSN_STATEMENT_CACHE_TABLE
                                     5. DSN_QUERYINFO_TABLE
                                     6. DSN_PREDICAT_TABLE
                                     7. DSN_USERQUERY_TABLE
                                     8. DSN_PREDICATE_SELECTIVITY
```

Note: The DPS and SCT options are displayed only if the subsystem is running on Db2 12 for z/OS, the CACHEDYN subsystem parameter is set to YES, and the CACHEDYN_STABILIZATION subsystem parameter is set to something other than NONE.

*Figure 338. **EXPLAIN (ADB2E)** panel*

3. On the **Explain an SQL Statement (ADB2EE)** panel, specify the following information, and press Enter to run the EXPLAIN statement:

   - Specify a query number and an SQL statement. If you leave the query number blank, Db2 Admin Tool generates a query number for you in the form *YYMMDDSSS*, where YYMMDD is the year/month/day and *SSS* is a sequence number.
   - **Optional:** Use the **SET CURRENT DEGREE** field to set the current degree of parallelism before running the EXPLAIN plan statement. Valid values are 1 and ANY. If the field is blank, the current degree is not changed.



```
 ADB2EE in ---------------- DD1A Explain an SQL Statement ---------------- 15:50
 Command
===>



 SET CURRENT DEGREE =       ;   (Optional)                DB2 System: DD1A
 EXPLAIN ALL                                              DB2 SQL ID: ADM001

 SET QUERYNO =
 Query number==>
 FOR
 SQL stmt    ==>
 SELECT * FROM SYSIBM.SYSTABLES WHERE NAME LIKE 'SYS';
















 Press ENTER to execute explain, or enter EDIT on the command line to edit
 the SQL statement.
```

4. On the **Rows from PLAN_TABLE (ADB2EL)** panel, use the I line command to display the EXPLAIN results.

Listing plan table rows

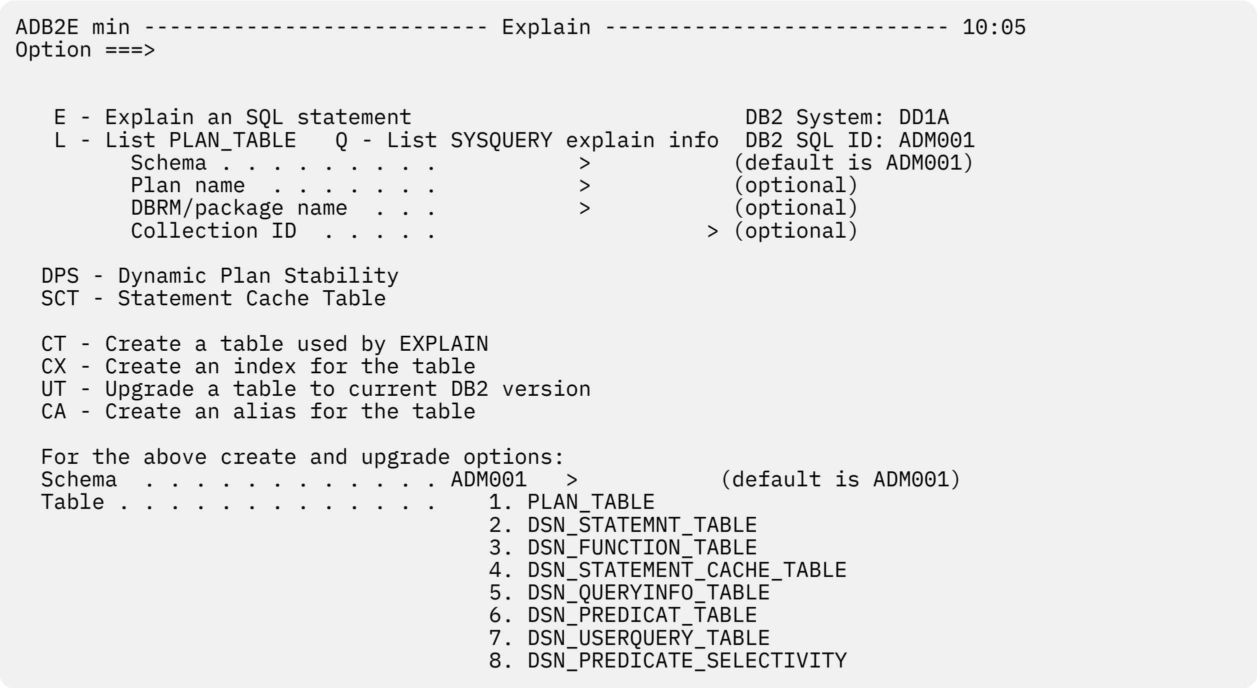
A plan table, PLAN_TABLE, is a Db2 EXPLAIN table that contains access path information for SQL statements.

Procedure

To view a plan table:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option E, and press Enter.
2. On the **EXPLAIN (ADB2E)** panel, specify option L and optionally any qualifiers in the following fields to identify which plan table you want to view:
   - **Schema**
   - **Plan name**
   - **DBRM/package name**
   - **Collection ID**

   If you do not specify any qualifiers, the default schema name listed on the panel is used.



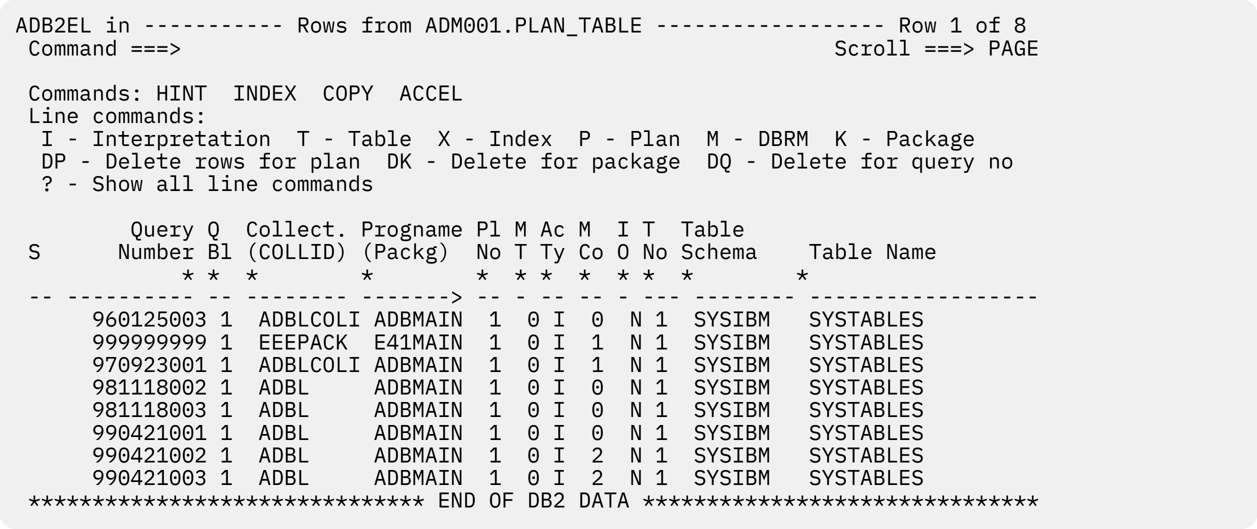
```
ADB2E min --------------------------- Explain --------------------------- 10:05
Option ===>


   E - Explain an SQL statement                           DB2 System: DD1A
   L - List PLAN_TABLE   Q - List SYSQUERY explain info  DB2 SQL ID: ADM001
         Schema . . . . . . . . .           >           (default is ADM001)
         Plan name  . . . . . . .           >           (optional)
         DBRM/package name  . . .           >           (optional)
         Collection ID  . . . . .                     > (optional)

  DPS - Dynamic Plan Stability
  SCT - Statement Cache Table

  CT - Create a table used by EXPLAIN
  CX - Create an index for the table
  UT - Upgrade a table to current DB2 version
  CA - Create an alias for the table

  For the above create and upgrade options:
  Schema  . . . . . . . . . . . . ADM001   >           (default is ADM001)
  Table . . . . . . . . . . . . .    1. PLAN_TABLE
                                     2. DSN_STATEMNT_TABLE
                                     3. DSN_FUNCTION_TABLE
                                     4. DSN_STATEMENT_CACHE_TABLE
                                     5. DSN_QUERYINFO_TABLE
                                     6. DSN_PREDICAT_TABLE
                                     7. DSN_USERQUERY_TABLE
                                     8. DSN_PREDICATE_SELECTIVITY
```

Note: The DPS and SCT options are displayed only if the subsystem is running on Db2 12 for z/OS, the CACHEDYN subsystem parameter is set to YES, and the CACHEDYN_STABILIZATION subsystem parameter is set to something other than NONE.

*Figure 339. **EXPLAIN (ADB2E)** panel*

3. Press Enter.

   The **Rows from PLAN_TABLE (ADB2EL)** panel is displayed:



```
ADB2EL in ----------- Rows from ADM001.PLAN_TABLE ------------------ Row 1 of 8
 Command ===>                                                    Scroll ===> PAGE

 Commands: HINT  INDEX  COPY  ACCEL
 Line commands:
  I - Interpretation  T - Table  X - Index  P - Plan  M - DBRM  K - Package
  DP - Delete rows for plan  DK - Delete for package  DQ - Delete for query no
  ? - Show all line commands

         Query Q  Collect. Progname Pl M Ac M  I T  Table
 S      Number Bl (COLLID) (Packg)  No T Ty Co O No Schema    Table Name
             * *  *        *        *  * *  *  * *  *        *
 -- ---------- -- -------- -------> -- - -- -- - --- -------- ------------------
      960125003 1  ADBLCOLI ADBMAIN  1  0 I  0  N 1  SYSIBM   SYSTABLES
      999999999 1  EEEPACK  E41MAIN  1  0 I  1  N 1  SYSIBM   SYSTABLES
      970923001 1  ADBLCOLI ADBMAIN  1  0 I  1  N 1  SYSIBM   SYSTABLES
      981118002 1  ADBL     ADBMAIN  1  0 I  0  N 1  SYSIBM   SYSTABLES
      981118003 1  ADBL     ADBMAIN  1  0 I  0  N 1  SYSIBM   SYSTABLES
      990421001 1  ADBL     ADBMAIN  1  0 I  0  N 1  SYSIBM   SYSTABLES
      990421002 1  ADBL     ADBMAIN  1  0 I  2  N 1  SYSIBM   SYSTABLES
      990421003 1  ADBL     ADBMAIN  1  0 I  2  N 1  SYSIBM   SYSTABLES
 ******************************* END OF DB2 DATA *******************************
```

*Figure 340. **Rows from PLAN_TABLE (ADB2EL)** panel*

   If this panel is not displayed, the plan table is empty.

   The release level and mode of your Db2 subsystem affect the options that are available.

   For more information about any of the columns on this panel, see the online help and PLAN_TABLE (Db2 12 for z/OS documentation) .

What to do next

Use this panel to see how Db2 will execute SQL statements. This information is gathered from previously executed EXPLAIN statements and from Db2 BIND commands that specify EXPLAIN(YES).

You can view the information on this panel in different formats. To switch between formats, use the following primary commands:

COL
Package mode, which shows Collection (COLLID) and Progname (PACKG)

HINT
Hint mode, which shows Hint ID and Hint Used

INDEX
Index information

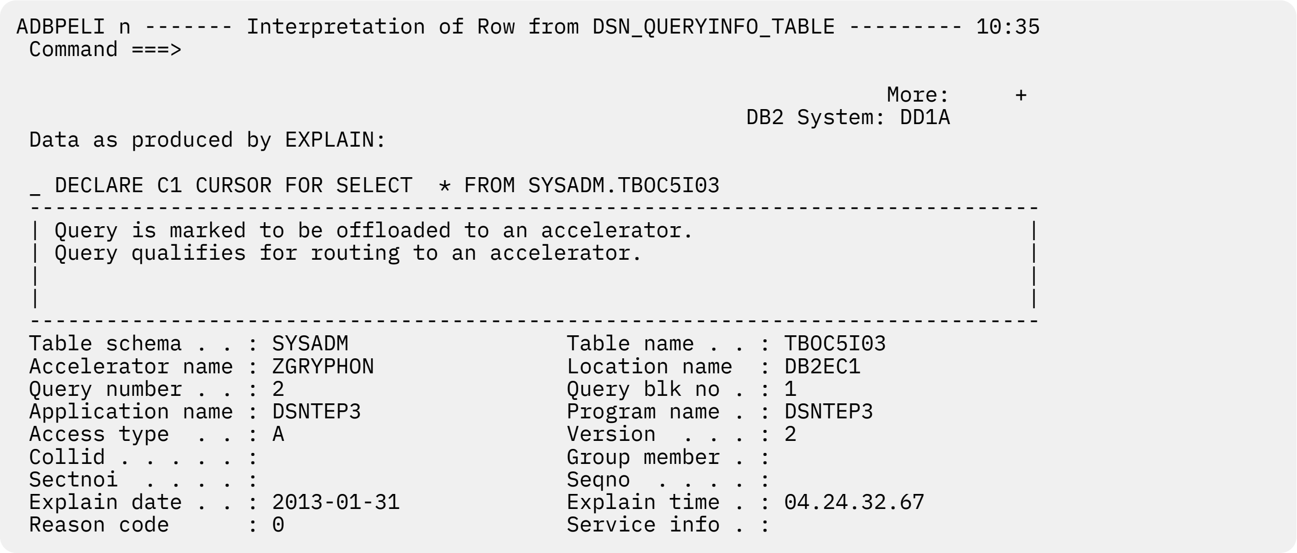
TABLE
Table information

COPY
Copy displayed rows to another PLAN_TABLE

ACCEL
Accelerator server information. This format is available only if you are using an accelerator.

To display the interpretation information for any queries, issue line command I.

For accelerated queries, I displays an interpretation panel similar to the panel in the following figure. Accelerated queries have an access type of A (accesstype = 'A').



```
ADBPELI n ------- Interpretation of Row from DSN_QUERYINFO_TABLE --------- 10:35
 Command ===>

                                                                    More:     +
                                                         DB2 System: DD1A
 Data as produced by EXPLAIN:

 _ DECLARE C1 CURSOR FOR SELECT  * FROM SYSADM.TBOC5I03
 -------------------------------------------------------------------------------
 | Query is marked to be offloaded to an accelerator.                          |
 | Query qualifies for routing to an accelerator.                              |
 |                                                                             |
 |                                                                             |
 -------------------------------------------------------------------------------
 Table schema . . : SYSADM                   Table name . . : TBOC5I03
 Accelerator name : ZGRYPHON                 Location name  : DB2EC1
 Query number . . : 2                        Query blk no . : 1
 Application name : DSNTEP3                  Program name . : DSNTEP3
 Access type  . . : A                        Version  . . . : 2
 Collid . . . . . :                          Group member . :
 Sectnoi  . . . . :                          Seqno  . . . . :
 Explain date . . : 2013-01-31               Explain time . : 04.24.32.67
 Reason code      : 0                        Service info . :
```

*Figure 341. **Interpretation of Row from DSN_QUERYINFO_TABLE (ADBPELI)** panel*

In addition to the EXPLAIN information, this panel states whether the query is marked to be offloaded to an accelerator and whether it is qualified to be routed to an accelerator. If the query is not qualified to be offloaded to an accelerator, the reason is stated on the panel.

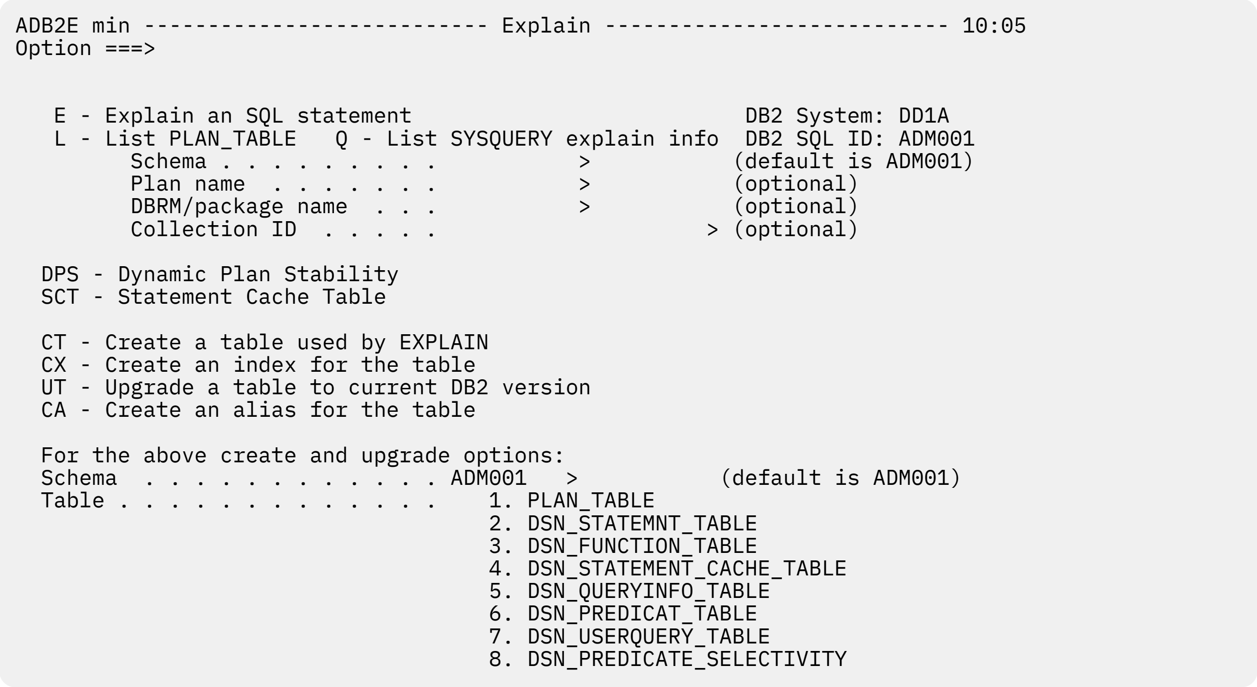
Creating EXPLAIN tables

Db2 EXPLAIN tables store the information that is captured when you run the SQL EXPLAIN statement. Before running EXPLAIN, you must create at least a plan table (PLAN_TABLE). If you want to use the EXPLAIN option STMTCACHE ALL, you must also create a statement cache table (DSN_STATEMENT_CACHE_TABLE). The other EXPLAIN tables are optional, depending on the information that you want to collect. For more information about these tables, see EXPLAIN tables (Db2 12 for z/OS documentation) .

Procedure

To create EXPLAIN tables:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option E, and press Enter.
2. On the **EXPLAIN (ADB2E)** panel, specify option CT on the command line and a number for the table in the **Table** field. Then, press Enter.



```
ADB2E min -------------------------- Explain -------------------------- 10:05
Option ===>


   E - Explain an SQL statement                          DB2 System: DD1A
   L - List PLAN_TABLE   Q - List SYSQUERY explain info  DB2 SQL ID: ADM001
         Schema . . . . . . . . .           >           (default is ADM001)
         Plan name  . . . . . . .           >           (optional)
         DBRM/package name  . . .           >           (optional)
         Collection ID  . . . . .                     > (optional)

  DPS - Dynamic Plan Stability
  SCT - Statement Cache Table

  CT - Create a table used by EXPLAIN
  CX - Create an index for the table
  UT - Upgrade a table to current DB2 version
  CA - Create an alias for the table

  For the above create and upgrade options:
  Schema  . . . . . . . . . . . . ADM001   >           (default is ADM001)
  Table . . . . . . . . . . . . .    1. PLAN_TABLE
                                     2. DSN_STATEMNT_TABLE
                                     3. DSN_FUNCTION_TABLE
                                     4. DSN_STATEMENT_CACHE_TABLE
                                     5. DSN_QUERYINFO_TABLE
                                     6. DSN_PREDICAT_TABLE
                                     7. DSN_USERQUERY_TABLE
                                     8. DSN_PREDICATE_SELECTIVITY
```

Note: The DPS and SCT options are displayed only if the subsystem is running on Db2 12 for z/OS, the CACHEDYN subsystem parameter is set to YES, and the CACHEDYN_STABILIZATION subsystem parameter is set to something other than NONE.

Figure 342. ***EXPLAIN (ADB2E)*** *panel*

3. On the **Create XXXXX_TABLE (ADB2EC)** panel, specify the desired values, and press Enter to create the table.

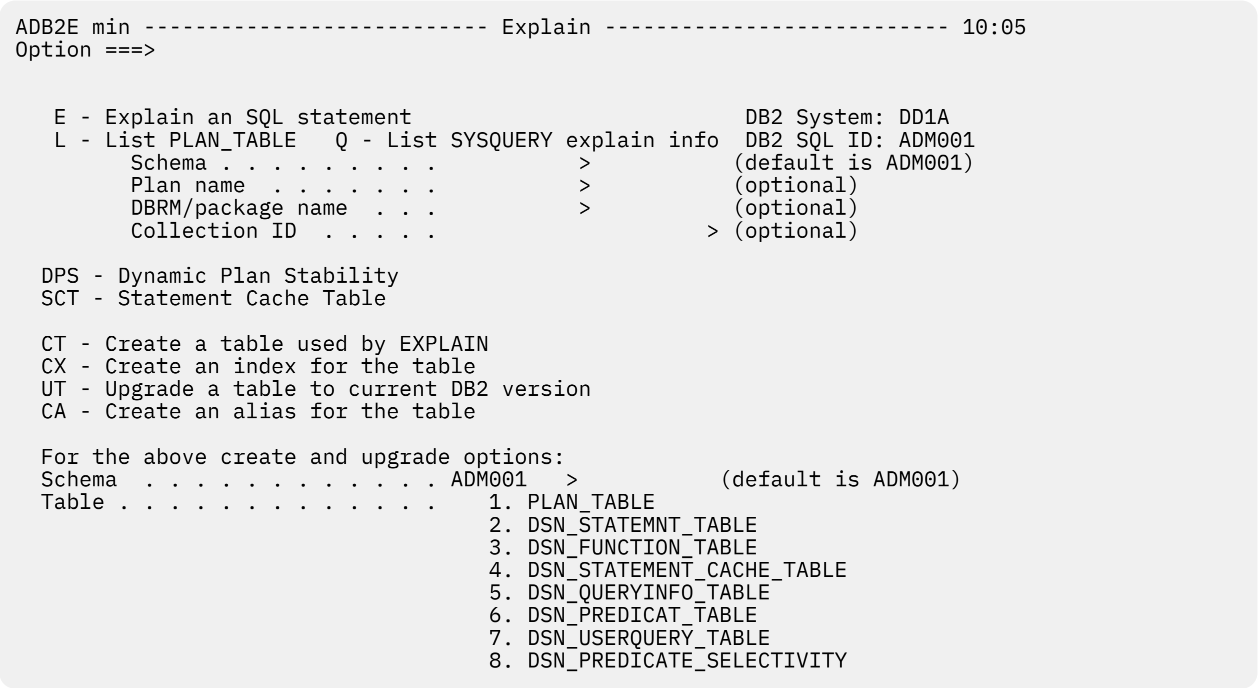
Upgrading EXPLAIN tables

The format of Db2 EXPLAIN tables can vary in different versions of Db2. Therefore, you might need to upgrade your EXPLAIN tables when you migrate to a new version of Db2 or when you apply certain Db2 maintenance.

Procedure

To upgrade EXPLAIN tables:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option E, and press Enter.
2. On the **EXPLAIN (ADB2E)** panel, specify option UT on the command line and a number for the table in the **Table** field. Then, press Enter.



```
ADB2E min -------------------------- Explain -------------------------- 10:05
Option ===>


   E - Explain an SQL statement                          DB2 System: DD1A
   L - List PLAN_TABLE   Q - List SYSQUERY explain info  DB2 SQL ID: ADM001
         Schema . . . . . . . . .           >           (default is ADM001)
         Plan name  . . . . . . .           >           (optional)
         DBRM/package name  . . .           >           (optional)
         Collection ID  . . . . .                     > (optional)

  DPS - Dynamic Plan Stability
  SCT - Statement Cache Table

  CT - Create a table used by EXPLAIN
  CX - Create an index for the table
  UT - Upgrade a table to current DB2 version
  CA - Create an alias for the table

  For the above create and upgrade options:
  Schema  . . . . . . . . . . . . ADM001   >           (default is ADM001)
  Table . . . . . . . . . . . . .    1. PLAN_TABLE
                                     2. DSN_STATEMNT_TABLE
                                     3. DSN_FUNCTION_TABLE
                                     4. DSN_STATEMENT_CACHE_TABLE
                                     5. DSN_QUERYINFO_TABLE
                                     6. DSN_PREDICAT_TABLE
                                     7. DSN_USERQUERY_TABLE
                                     8. DSN_PREDICATE_SELECTIVITY
```

Note: The DPS and SCT options are displayed only if the subsystem is running on Db2 12 for z/OS, the CACHEDYN subsystem parameter is set to YES, and the CACHEDYN_STABILIZATION subsystem parameter is set to something other than NONE.

Figure 343. ***EXPLAIN (ADB2E)*** *panel*

Db2 Admin Tool issues a series of ALTER TABLE statements to upgrade the table so that it contains the maximum number of columns that is supported by the current version of Db2. A message indicates whether the table is upgraded successfully.

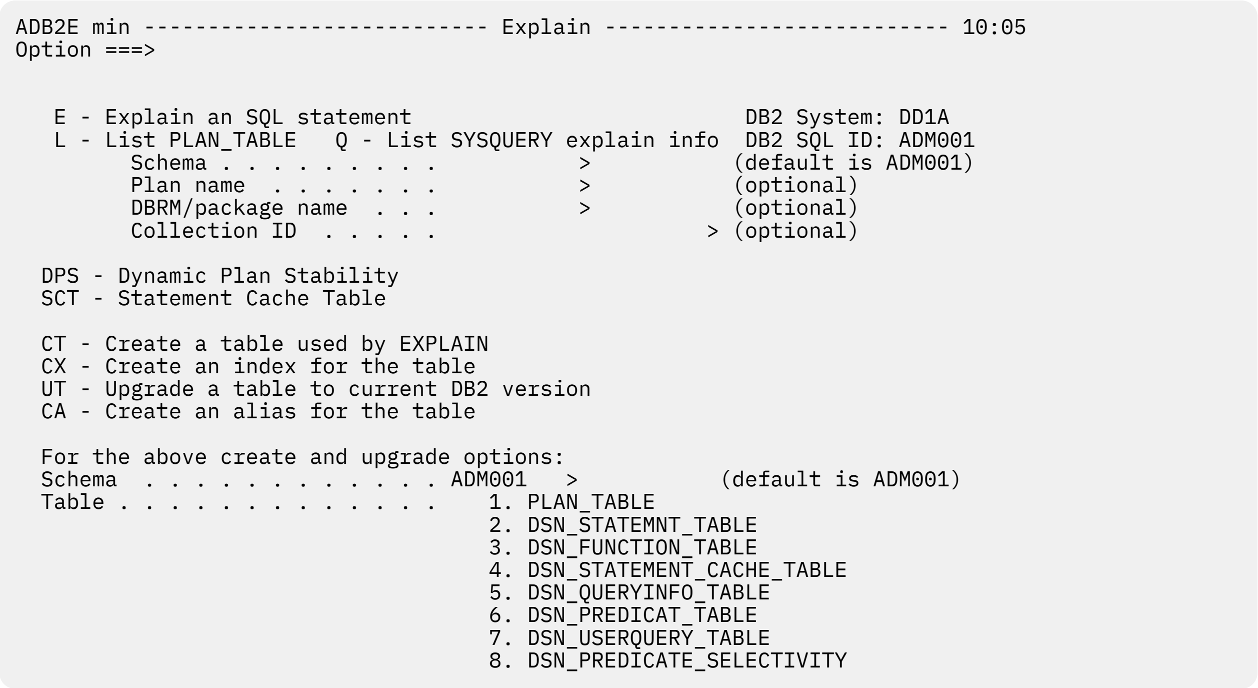
Creating aliases for EXPLAIN tables

An alias allows a user with SELECT and INSERT privileges to populate EXPLAIN tables that are created under a different AUTH ID.

Procedure

To create aliases for EXPLAIN tables:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option E, and press Enter.
2. On the **EXPLAIN (ADB2E)** panel, specify option CA on the command line and a number for the table in the **Table** field. Then, press Enter.

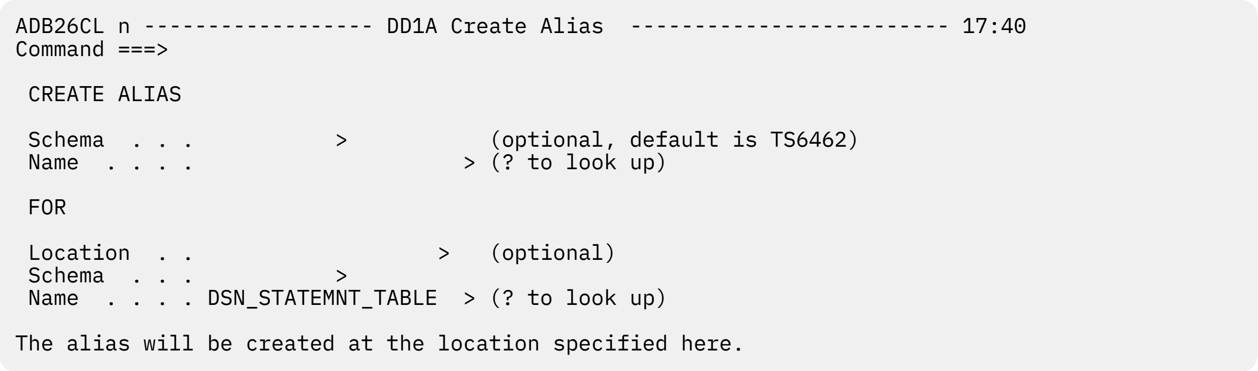


```
ADB2E min -------------------------- Explain -------------------------- 10:05
Option ===>


   E - Explain an SQL statement                          DB2 System: DD1A
   L - List PLAN_TABLE   Q - List SYSQUERY explain info  DB2 SQL ID: ADM001
         Schema . . . . . . . . .           >           (default is ADM001)
         Plan name  . . . . . . .           >           (optional)
         DBRM/package name  . . .           >           (optional)
         Collection ID  . . . . .                     > (optional)

  DPS - Dynamic Plan Stability
  SCT - Statement Cache Table

  CT - Create a table used by EXPLAIN
  CX - Create an index for the table
  UT - Upgrade a table to current DB2 version
  CA - Create an alias for the table

  For the above create and upgrade options:
  Schema  . . . . . . . . . . . . . ADM001   >           (default is ADM001)
  Table . . . . . . . . . . . . . .    1. PLAN_TABLE
                                       2. DSN_STATEMNT_TABLE
                                       3. DSN_FUNCTION_TABLE
                                       4. DSN_STATEMENT_CACHE_TABLE
                                       5. DSN_QUERYINFO_TABLE
                                       6. DSN_PREDICAT_TABLE
                                       7. DSN_USERQUERY_TABLE
                                       8. DSN_PREDICATE_SELECTIVITY
```

Note: The DPS and SCT options are displayed only if the subsystem is running on Db2 12 for z/OS, the CACHEDYN subsystem parameter is set to YES, and the CACHEDYN_STABILIZATION subsystem parameter is set to something other than NONE.

*Figure 344. **EXPLAIN (ADB2E)** panel*

3. On the **Create Alias (ADB26CL)** panel, specify the alias and options that you want, and press Enter.



```
ADB26CL n ------------------ DD1A Create Alias  ------------------------- 17:40
Command ===>

 CREATE ALIAS

 Schema  . . .           >           (optional, default is TS6462)
 Name  . . . .                     > (? to look up)

 FOR

 Location  . .                   >   (optional)
 Schema  . . .           >
 Name  . . . . DSN_STATEMNT_TABLE  > (? to look up)

The alias will be created at the location specified here.
```

*Figure 345. **Create Alias (ADB26CL)** panel*

After the CREATE ALIAS statement completes successfully, the alias is created.

Copying a plan table

You can copy rows from one plan table (PLAN_TABLE) to another plan table with a different schema by using Db2 Admin Tool.

Procedure

To copy a plan table:

1. View the plan table that you want to copy.
2. On the **Rows from PLAN_TABLE (ADB2EL)** panel, take one of the following actions:
   - To copy all rows displayed, issue the COPY primary command, and press Enter.

You can first filter the list of rows displayed by using the search argument area or the SARG command; see “Filtering data on a panel after the query result is returned” on page 200.

- To copy all rows with a specific query number, issue the CQ line command against one of those rows, and press Enter:

```
ADB2EL in ----------- Rows from J148286.PLAN_TABLE   ------ Row 1 to 41 of 137
Command ===>                                                  Scroll ===> PAGE

Commands: HINT  INDEX  COPY  ACCEL
Line commands:
 I - Interpretation  T - Table  X - Index  P - Plan  M - DBRM  K - Package
 DP - Delete rows for plan  DK - Delete for package  DQ - Delete for query no
 ? - Show all line commands

        Query Q  Collect. Progname Pl M Ac M  I  T  Table
S      Number Bl (COLLID) (Packg)  No T Ty Co O  No Schema   Table Name
            * *  *        *        *  * *  *  * *   *        *
-- ---------- -- -------- -------- -- - -- -- - --- -------- ----------------->
CQ       1133 1  JWR      ADB0     1  0 I  2  N   1 SYSIBM   SYSTABLES
         1227 1  JWR      ADB0     1  0 I  2  N   1 SYSACCEL SYSACCELERATEDTABL
         1338 1  JWR      ADB0     1  0 I  0  N   1 SYSIBM   SYSPACKAGE
         1338 1  JWR      ADB0     1  0 I  2  Y   1 SYSIBM   SYSTABLES
```

*Figure 346. **Rows from PLAN_TABLE (ADB2EL)** panel*

3. On the **Copy entries (ADBPELC)** panel, specify the appropriate options, and press Enter to copy the rows:

```
ADBPELC n ----------------------- DD1A Copy entries ---------------- 15:24
Command ===>

 Copy contents from <schema>.PLAN_TABLE                                  >
   To schema  . . . . . . . . .             >
   To table name  . . . . . . .                    > (Default PLAN_TABLE)
   Delete rows prior to copy  .           (A - All, M - Matching, N - None)

 Show this panel prior to each use .      (Yes/No)
```

Depending the value of the **Show this panel prior to each use** field, this panel is displayed when the first of one or more rows are chosen on the **Rows from PLAN_TABLE (ADB2EL)** panel. Use this field to simplify copying multiple individual rows to the same target table without asking for the target information for each row. The option is reset to blank each time the **Rows from PLAN_TABLE (ADB2EL)** panel is first displayed.

The **Delete rows prior to copy** action is done after the panel is displayed. If the value of the **Show this panel prior to each use** is changed to NO, the **Delete rows prior to copy** action is done once. If the **Show this panel prior to each use** is set to YES, the **Delete rows prior to copy** action is performed each time that the panel is displayed. In this case, the value of **Delete rows prior to copy** should be set to All when the panel is first displayed, and then set to None for subsequent panels so that the newly copied row from the first display is retained.

Stabilizing dynamic SQL statements

You can use the *dynamic SQL plan stability* feature of Db2 to help achieve access path stability for dynamic SQL statements. When this feature is enabled, Db2 stores and retrieves statement cache structures in the catalog to avoid full prepare operations.

Before you begin

Before you can stabilize dynamic SQL statements, all of the following conditions must be true:

- The subsystem must be running on Db2 12 function level 500 or higher.
- The CACHEDYN subsystem parameter must be set to YES, and the CACHEDYN_STABILIZATION subsystem parameter must be set to either BOTH or CAPTURE. (To change these parameter values, follow the instructions in “Managing Db2 subsystem parameters” on page 971.)

Procedure

To stabilize dynamic SQL statements:

1. Identify dynamic SQL statements to stabilize.
2. Capture dynamic SQL statements for stabilization.
3. Optional: View active dynamic query capture monitors.
4. Optional: View stabilized dynamic SQL statements.
5. Optional: Remove stabilized SQL statements from the catalog.

Related information

Dynamic SQL plan stability (Db2 12 for z/OS)
CACHEDYN subsystem parameter (Db2 12 for z/OS)
CACHEDYN_STABILIZATION subsystem parameter (Db2 12 for z/OS)

Identifying dynamic SQL statements to stabilize

You can examine the dynamic statement cache to determine SQL statements that are likely to be captured and stabilized when you use the dynamic SQL plan stability feature of Db2. To examine the statement cache, you can view the statement cache table, DSN_STATEMENT_CACHE_TABLE. This table captures information about statements in the statement cache when an EXPLAIN STATEMENT CACHE ALL statement is issued.

Before you begin

Before you can view the statement cache table, all of the following conditions must be true:

- The subsystem must be running on Db2 12 function level 500 or higher.
- The CACHEDYN subsystem parameter must be set to YES, and the CACHEDYN_STABILIZATION subsystem parameter must be set to either BOTH or CAPTURE. (To change these parameter values, follow the instructions in "Managing Db2 subsystem parameters" on page 971.)
- The DSN_STATEMENT_CACHE_TABLE must exist and be populated. (For instructions on how to create this table, see "Creating EXPLAIN tables" on page 618. For instructions on how to populate this table, see "Explaining SQL Statements" on page 614.)

Procedure

To identify dynamic SQL statements to stabilize:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option E, and press Enter.
2. On the **EXPLAIN (ADB2E)** panel, specify option SCT, and press Enter.

   If the SCT option is not displayed, the prerequisites (in Before you begin) are not all true.
3. If a performance trace with IFCIDs 0316, 0317, and 0318 is not active, start this trace:

   a) On the **Display/Manage Statement Cache Table (ADB2ES)** panel, specify option 1.

   b) On the **Display/Manage Traces (ADB2Z2T2)** panel, specify the STA command:

```
ADB2Z2T2  ------------- VC1A Display/Manage Traces ----------- Row 1 to 4 of 4
 Command ===> STA                                              Scroll ===> CSR

 Commands: STA - Start trace
 Line commands: STA - Start trace  STO - Stop trace  DIS - Display trace details
   MT - Modify trace

     T  Trace
 Sel No Type   Trace Classes               Dest Qual IFCID
     *  *      *                           *    *    *
--- -- ------ --------------------------- ---- ---- ----------------------
     01 STAT   01,03,04                   SMF  NO
     02 ACCTG  01,02,03                   SMF  NO
     03 MON    01                         OP1  NO
 ******************************* END OF DB2 DATA *******************************
```

The **Trace Functions (ADB2Z2TS)** panel opens with the required performance trace specified:

```
ADB2Z2TS  -------------------- DC1A Trace Functions   ------------------- 11:12
Command ===>

                                                                  More:     +
-START

 TRACE . . . . . . . . . PERFM       (STat, ACctg, AUdit, PErfm or MOnitor)
 CLASS . . . . . . . . . 30                                                 >
 DEST  . . . . . . . . . OPX       > (SMF, GTF, OPn, OPX and/or SRV)
 SCOPE . . . . . . . . .             (L - Local, G - Group)
 IFCID . . . . . . . . . 316,317,318                                        >
 BUFSIZE . . . . . . . .             (8-1024)

 TDATA CORRELATION
  Include cor header . .             (Yes/No)
  Include CPU header . .             (Yes/No)
  Include trace hdr  . .             (Yes/No)
  Include dist hdr . . .             (Yes/No)
 COMMENT . . . . . . . .
 RMID  . . . . . . . . .
 AUDTPLCY  . . . . . . .
```

c) Press Enter to start the trace.

d) If the **Statement Execution Prompt (ADB2PSTM)** panel opens, follow the instructions on the panel to run the command to start the trace.

e) Exit back to **Display/Manage Statement Cache Table (ADB2ES)** panel

4. Optional: On the **Display/Manage Statement Cache Table (ADB2ES)** panel, specify a schema and any other criteria to filter the DSN_STATEMENT_CACHE_TABLE rows that you want to view:

```
ADB2ES in ----------------- VC1A Statement Cache Table ----------------- 17:32
Option ===>

 1 - Display/manage traces
 2 - Display/manage DSN_STATEMENT_CACHE_TABLE
       Schema . . . TS5771 > (Default is TS5771)

Enter display selection criteria for Option 2.
Settings: LIKE operator; Criteria not saved.

Stabilization group .                                                   >
SQL ID  . . . . . . .                                                   >
Literal replaced  . .          (R,D,or blank)
Stabilized  . . . . .          (Yes/No)
SQL statement type  .          (Select,With,Update,Insert,Delete,Merge or etc)
Elapsed time => . . .
CPU time => . . . . .
Execution count =>  .
Cached within . . . .
Populated within  . .
```

5. On the **Display/Manage Statement Cache Table (ADB2ES)** panel, specify option 2, and press Enter.

   The **Rows from DSN_STATEMENT_CACHE_TABLE (ADB2ES2)** panel shows the contents of the statement cache table:

```
ADB2ES2 n --- DC1A Rows from TS5771.DSN_STATEMENT_CACHE_T Row 1 to 13 of 1,000
Command ===>                                                  Scroll ===> CSR
                                                                     More:    >
Commands: CLEAR  EXPLSTMTCACHE  CAPTURE  STARTTRACE
Line commands:
 SID - STA DYNQUERY with STMT_ID  STK - STA DYNQUERY with STMT_TOKEN
 ? - Show all line commands

Stabilization group . . SG1                                                  >

              Prog
S     Stmt ID Name     SQL Id   S SQL Stmt
            * *        *        * *
--- <-------- -------- -------- - -------------------------------------------->
       948396 DSN@EP2L TS6025   N INSERT INTO RSTEST.STAFF_51174 VALUES( 104753
       957211 DSN@EP2L TS6025   N INSERT INTO RSTEST.STAFF_51174 VALUES( 113568
       958636 DSN@EP2L TS6025   N INSERT INTO RSTEST.STAFF_51174 VALUES( 114993
       958836 DSN@EP2L TS6025   N INSERT INTO RSTEST.STAFF_51174 VALUES( 115193
```

DSN_STATEMENT_CACHE_TABLE contains a snapshot of data from every time that the EXPLSTMTCACHE command is executed (or the Db2 EXPLAIN STATEMENT CACHE ALL statement is issued). Therefore, you might see some redundant entries. The table is not cleared until you issue the CLEAR command.

6. Examine the data in the table to determine which statements you want to stabilize:

   You can scroll horizontally to see additional columns in DSN_STATEMENT_CACHE_TABLE.

   If you want to update the EXPLAIN data in this table, issue the CLEAR command and then the EXPLSTMTCACHE command.

7. Capture dynamic SQL statements by completing one of the following actions:

   - To capture a single statement, use the SID or STK line commands.
   - To capture a group of statements or specify a threshold value, specify the CAPTURE command and follow the instructions in "Capturing dynamic SQL statements for stabilization" on page 625, starting at step "4" on page 626.

Related information

DSN_STATEMENT_CACHE_TABLE (Db2 12 for z/OS)

CACHEDYN subsystem parameter (Db2 12 for z/OS)

CACHEDYN_STABILIZATION subsystem parameter (Db2 12 for z/OS)

Capturing dynamic SQL statements for stabilization

As part of the dynamic SQL plan stability feature of Db2, you can request that Db2 capture dynamic SQL statements from the statement cache and store these statement cache structures in the Db2 catalog. This action can help achieve access path stability.

Before you begin

Before you can use Db2 Admin Tool to capture dynamic SQL statements for stabilization, all of the following conditions must be true:

- The subsystem must be running on Db2 12 function level 500 or higher.
- The CACHEDYN subsystem parameter must be set to YES, and the CACHEDYN_STABILIZATION subsystem parameter must be set to either BOTH or CAPTURE. (To change these parameter values, follow the instructions in "Managing Db2 subsystem parameters" on page 971.)

Additionally, if you want to specify a threshold value for stabilizing SQL statements, a performance trace with IFCIDs 0316 and 0318 must be active. (For instructions on how to start a trace, see "Identifying dynamic SQL statements to stabilize " on page 623 and follow the instructions up to step "3" on page 623.)

Procedure

To capture dynamic SQL statements for stabilization:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option E, and press Enter.
2. On the **EXPLAIN (ADB2E)** panel, specify option DPS, and press Enter.

   If the DPS option is not displayed, the prerequisites (in Before you begin) are not all true.
3. On the **Display/Manage Dynamic Plan Stability (ADB2ED)** panel, specify option 1, and press Enter:

```
ADB2ED in ---------- DC1A Display/Manage Dynamic Plan Stability --------- 14:35
Option ===> 1

OPTIONS:
  1 - START DYNQUERYCAPTURE
  2 - DISPLAY/STOP DYNQUERYCAPTURE
  3 - Display/manage SYSDYNQRY


Enter standard selection criteria for Option 3.
Settings: LIKE operator; Criteria not saved.

Stabilization Group . .                                                      >
Copy ID . . . . . . . .    (0 or 4)
SQL ID  . . . . . . . .
Valid . . . . . . . . .    (A,H,N or Y)
SQL Statement Type  . .
Applcompat  . . . . . .           (eg. V12R1M501)
Last Used . . . . . . .
Altered Within  . . . .
```

4. On the **START DYNQUERYCAPTURE (ADB2ED1)** panel, specify a stabilization group name and any other options to indicate the statements that you want to stabilize, and press Enter:

```
ADB2ED1 n ------------------ DC1A START DYNQUERYCAPTURE ----------------- 14:49
Command ===>

 -START DYNQUERYCAPTURE
  Stabilization group  . .                                                   >

  Options:
   Threshold . . . . . . .
   SQL ID  . . . . . . . .                                                   >
   Monitor . . . . . . . .              (Yes/No)
   Scope . . . . . . . . .              (Local,Group)

  Statement ID . . . . . .
  Statement token  . . . .                                                   >
```

   The fields on this panel correspond to options of the Db2 command START DYNQUERYCAPTURE, which is issued when you press Enter. The first set of options (**Threshold**, **SQL ID**, **Monitor** and **Scope**), **Statement ID**, and **Statement Token** are mutually exclusive.

   To capture a particular statement or statements, specify values for the **Statement ID** field or the **Statement token** field.

   To start ongoing monitoring of cached statements, specify YES in the **Monitor** field. Otherwise, Db2 captures only an immediate snapshot of cached statements.
5. If the **Statement Execution Prompt (ADB2PSTM)** panel opens, follow the instructions on the panel to run the START DYNQUERYCAPTURE command.

Related information

Dynamic SQL plan stability (Db2 12 for z/OS)

CACHEDYN_STABILIZATION subsystem parameter (Db2 12 for z/OS)

CACHEDYN subsystem parameter (Db2 12 for z/OS)

START DYNQUERYCAPTURE command (Db2 12 for z/OS)

Viewing dynamic query capture monitors

Dynamic query capture monitors are used by Db2 to monitor dynamic SQL statements for stabilization. These monitors are part of the dynamic SQL plan stability feature of Db2.

Before you begin

Before you can view dynamic query capture monitors, all of the following conditions must be true:

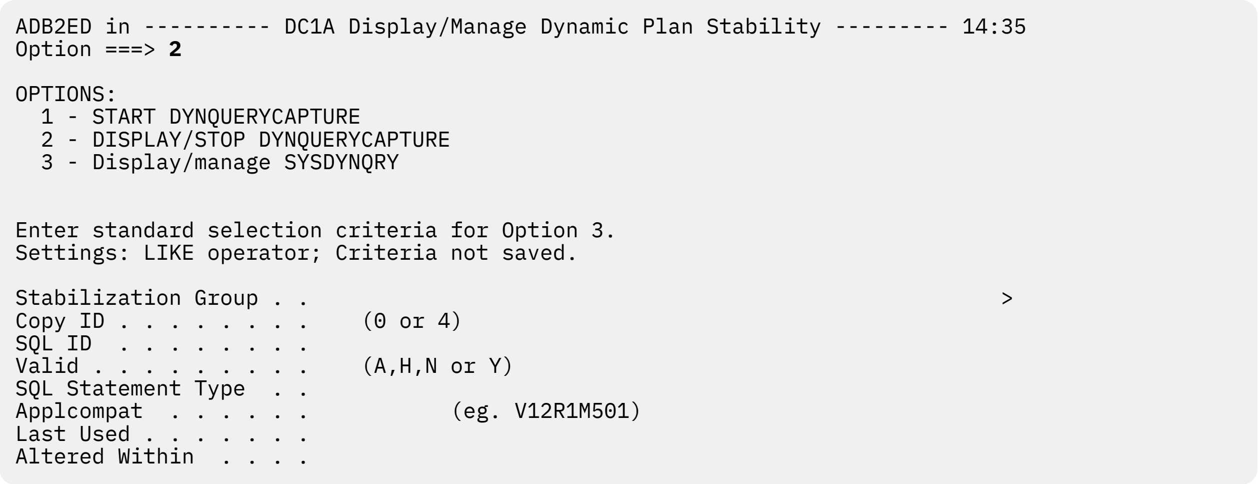
- The subsystem must be running on Db2 12 function level 500 or higher.
- The CACHEDYN subsystem parameter must be set to YES, and the CACHEDYN_STABILIZATION subsystem parameter must be set to either BOTH or CAPTURE. (To change these parameter values, follow the instructions in "Managing Db2 subsystem parameters" on page 971.)

Procedure

To view dynamic query capture monitors:

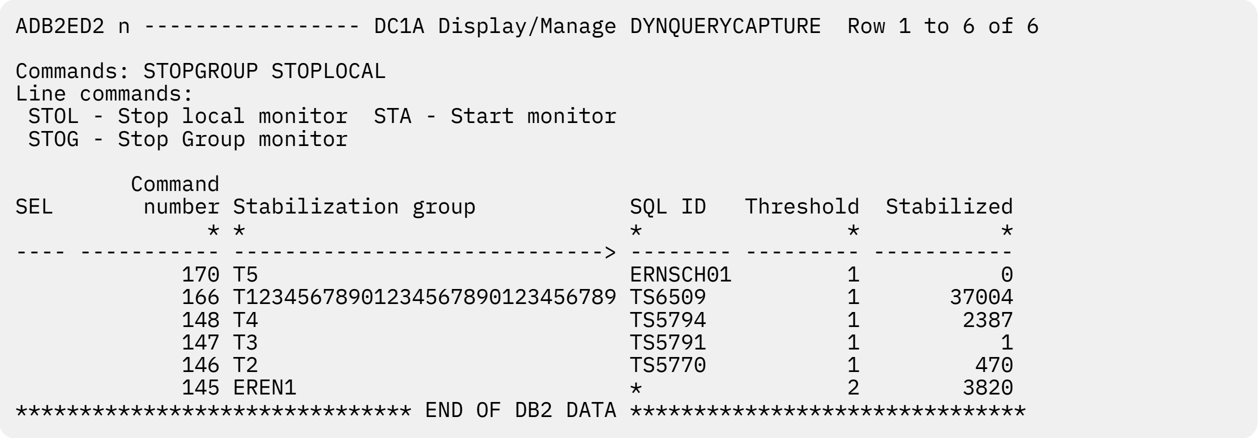
1. On the **DB2 Administration Menu (ADB2)** panel, specify option E, and press Enter.
2. On the **EXPLAIN (ADB2E)** panel, specify option DPS, and press Enter.

   If the DPS option is not displayed, the prerequisites (in Before you begin) are not all true.
3. Optional: On the **Display/Manage Dynamic Plan Stability (ADB2ED)** panel, specify any selection criteria to filter the monitors that you want displayed.
4. Specify option 2, and press Enter:



```
ADB2ED in ---------- DC1A Display/Manage Dynamic Plan Stability --------- 14:35
Option ===> 2

OPTIONS:
  1 - START DYNQUERYCAPTURE
  2 - DISPLAY/STOP DYNQUERYCAPTURE
  3 - Display/manage SYSDYNQRY


Enter standard selection criteria for Option 3.
Settings: LIKE operator; Criteria not saved.

Stabilization Group . .                                                      >
Copy ID . . . . . . . .    (0 or 4)
SQL ID  . . . . . . . .
Valid . . . . . . . . .    (A,H,N or Y)
SQL Statement Type  . .
Applcompat  . . . . . .           (eg. V12R1M501)
Last Used . . . . . . .
Altered Within  . . . .
```

The **Display/Manage DYNQUERYCAPTURE (ADB2ED2)** panel displays information about all of the currently active dynamic query capture monitors. Each row contains information about a stabilization group that is being monitored. If you started a dynamic query capture monitor with a smaller scope than an existing active query capture monitor, the monitor with the smaller scope is not considered active and is not displayed on this panel.



```
ADB2ED2 n ----------------- DC1A Display/Manage DYNQUERYCAPTURE  Row 1 to 6 of 6

Commands: STOPGROUP STOPLOCAL
Line commands:
 STOL - Stop local monitor  STA - Start monitor
 STOG - Stop Group monitor

         Command
SEL       number Stabilization group            SQL ID   Threshold  Stabilized
               * *                              *                 *          *
---- ----------- ------------------------------> -------- --------- -----------
             170 T5                             ERNSCH01          1          0
             166 T12345678901234567890123456789 TS6509            1      37004
             148 T4                             TS5794            1       2387
             147 T3                             TS5791            1          1
             146 T2                             TS5770            1        470
             145 EREN1                          *                 2       3820
******************************* END OF DB2 DATA ********************************
```

From this panel, you can stop monitors by using the STOPGROUP and STOPLOCAL commands or the STOL and STOG line commands. You can also modify a monitor by using the STA line command or start a new monitor by using the STA command.

Related tasks

“Capturing dynamic SQL statements for stabilization” on page 625
As part of the dynamic SQL plan stability feature of Db2, you can request that Db2 capture dynamic SQL statements from the statement cache and store these statement cache structures in the Db2 catalog. This action can help achieve access path stability.

Related information

Dynamic SQL plan stability (Db2 12 for z/OS)

CACHEDYN_STABILIZATION subsystem parameter (Db2 12 for z/OS)

CACHEDYN subsystem parameter (Db2 12 for z/OS)

Viewing stabilized dynamic SQL statements

If you are using the dynamic SQL plan stability feature of Db2, you can view information about access path stabilization for captured dynamic SQL statements. This information is stored in the SYSDYNQRY catalog table.

Before you begin

Before you can view stabilization information, all of the following conditions must be true:

- The subsystem must be running on Db2 12 function level 500 or higher.
- The CACHEDYN subsystem parameter must be set to YES, and the CACHEDYN_STABILIZATION subsystem parameter must be set to either BOTH or CAPTURE. (To change these parameter values, follow the instructions in “Managing Db2 subsystem parameters” on page 971.)
- You must have started capturing dynamic SQL statements for stabilization.

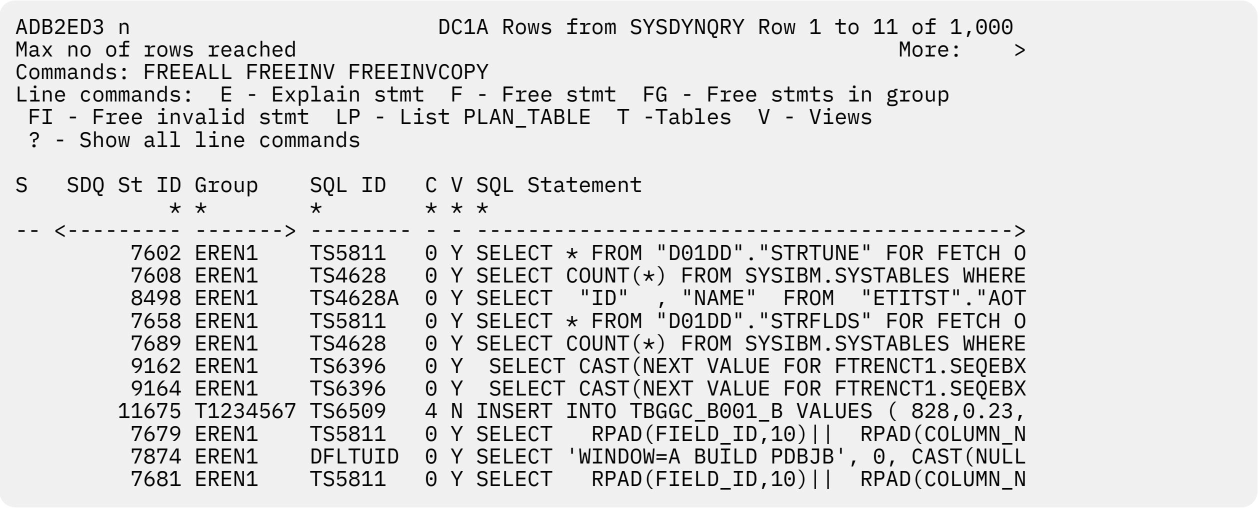
Procedure

To view stabilized dynamic SQL statements:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option E, and press Enter.
2. On the **EXPLAIN (ADB2E)** panel, specify option DPS, and press Enter.

   If the DPS option is not displayed, the prerequisites (in Before you begin) are not all true.
3. On the **Display/Manage Dynamic Plan Stability (ADB2ED)** panel, specify option 3, and press Enter.

   The **Rows from SYSDYNQRY (ADB2ED3)** panel shows information about all of the statements that have been captured:



```
ADB2ED3 n                        DC1A Rows from SYSDYNQRY Row 1 to 11 of 1,000
Max no of rows reached                                               More:    >
Commands: FREEALL FREEINV FREEINVCOPY
Line commands:  E - Explain stmt  F - Free stmt  FG - Free stmts in group
 FI - Free invalid stmt  LP - List PLAN_TABLE  T -Tables  V - Views
 ? - Show all line commands

S   SDQ St ID Group    SQL ID   C V SQL Statement
            * *        *        * * *
-- <--------- -------> -------- - - ------------------------------------------->
         7602 EREN1    TS5811   0 Y SELECT * FROM "D01DD"."STRTUNE" FOR FETCH O
         7608 EREN1    TS4628   0 Y SELECT COUNT(*) FROM SYSIBM.SYSTABLES WHERE
         8498 EREN1    TS4628A  0 Y SELECT  "ID"  , "NAME"  FROM  "ETITST"."AOT
         7658 EREN1    TS5811   0 Y SELECT * FROM "D01DD"."STRFLDS" FOR FETCH O
         7689 EREN1    TS4628   0 Y SELECT COUNT(*) FROM SYSIBM.SYSTABLES WHERE
         9162 EREN1    TS6396   0 Y  SELECT CAST(NEXT VALUE FOR FTRENCT1.SEQEBX
         9164 EREN1    TS6396   0 Y  SELECT CAST(NEXT VALUE FOR FTRENCT1.SEQEBX
        11675 T1234567 TS6509   4 N INSERT INTO TBGGC_B001_B VALUES ( 828,0.23,
         7679 EREN1    TS5811   0 Y SELECT   RPAD(FIELD_ID,10)||  RPAD(COLUMN_N
         7874 EREN1    DFLTUID  0 Y SELECT 'WINDOW=A BUILD PDBJB', 0, CAST(NULL
         7681 EREN1    TS5811   0 Y SELECT   RPAD(FIELD_ID,10)||  RPAD(COLUMN_N
```

You can compare rows for the same query. If the **C** column contains 0, the copy is current. If the **C** column contains 4, the copy is invalid. If the **V** column contains Y, the stabilized access path for the dynamic query is valid.

From this panel, you can also determine more information about the statements by using the following line commands:

E
Run the EXPLAIN statement on the selected SQL statement.

T
Show the tables that are associated with the SQL statement.

V
Show the views that are associated with the SQL statement.

X
Show the indexes that are associated with the SQL statement.

S
Show the table spaces that are associated with the SQL statement.

DP
Show all object dependencies for the SQL statement.

LP
Show the plan table, which includes access path information, for the selected statement.

Related information

SYSDYNQRY catalog table (Db2 12 for z/OS)
CACHEDYN_STABILIZATION subsystem parameter (Db2 12 for z/OS)
CACHEDYN subsystem parameter (Db2 12 for z/OS)

Removing stabilized SQL statements from the catalog

If you captured dynamic SQL statements for stabilization, you can subsequently remove one or more stabilized dynamic queries from certain catalog tables. You might want to do so if the stabilized dynamic SQL statement is invalidated.

Before you begin

You must have started capturing dynamic SQL statements for stabilization.

Procedure

To remove stabilized SQL statements:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option E, and press Enter.
2. On the **EXPLAIN (ADB2E)** panel, specify option DPS, and press Enter.

   If the DPS option is not displayed, dynamic SQL plan stability is not enabled.
3. On the **Display/Manage Dynamic Plan Stability (ADB2ED)** panel, specify option 3, and press Enter.
4. On the **Rows from SYSDYNQRY (ADB2ED3)** panel, specify one of the following commands or line commands to free one or more stabilized queries:

   FREEALL
   Free all stabilized dynamic queries.

   FREEINV
   Free all invalid stabilized dynamic queries. (These queries have N in the **V** column to indicate that the stabilized access path is not valid.)

   FREEINVCOPY
   Free only copies of stabilized dynamic queries that are invalid. (These queries have 4 in the **C** column to indicate that the copy is invalid.)

   F
   Free the specified stabilized dynamic query.

   FG
   Free all statements in the stabilization group.

   FI
   Free any invalid statements with the same identifier. This command frees any statement that has the same value in the **SDQ St ID** column as the selected statement and N in the **V** column.

   Note: FREEALL, FREEINV, and FREEINVCOPY can also affect stabilized dynamic queries that are not displayed on the **Rows from SYSDYNQRY (ADB2ED3)** panel. For example, if this panel displays only stabilized dynamic queries for a specific statement ID, the FREEALL command frees all stabilized dynamic queries; the command is not limited to the stabilized dynamic queries that are displayed for a specific statement.

   The DSN subcommand FREE STABILIZED DYNAMIC QUERY is run for the statement or statements that you specified.
5. If the **Statement Execution Prompt (ADB2PSTM)** panel opens, follow the instructions on the panel to run the FREE STABILIZED DYNAMIC QUERY command.

Related information

FREE STABILIZED DYNAMIC QUERY (DSN) (Db2 12 for z/OS)

Space management

Db2 Admin Tool manages space by displaying Db2 and VSAM statistics for Db2 page sets and by invoking functions against objects.

Using the Db2 Admin Tool Space Manager panels, you can:

- Display Db2 and VSAM information about Db2 page sets and invoke functions against objects. The statistical data is gathered from the Db2 catalog and merged with data from the VSAM catalogs.
- Alter page set properties
- Resize page sets to eliminate extents and free unused space
- Change to and from STOGROUP- and VCAT-defined space
- Estimate primary and secondary space allocation for new table spaces or indexes

Restriction: The following limitations apply to the Db2 Admin Tool Space Manager:

- The resize function generates separate jobs for each page set that exceeds the limits specified (primary command RESZ). This means that an index is reorganized twice, first by reorganizing the table space and then by reorganizing the index if the criteria for resizing are met by both spaces. Only the specific job for the index will update the allocations for the index.

- Resize calculations are based on the High Used RBA for the VSAM data set that contains the table space or index. This means that if activity on tables has left freespace in the pages, resize might overallocate space. This can be verified by repeating the resize. Db2 Admin Tool Space Manager displays the message “No changes” if all selected spaces conform to the limitations given (number of extents, % used).

Some of the space management functions are available as REST APIs. See “Provided REST APIs” on page 876.

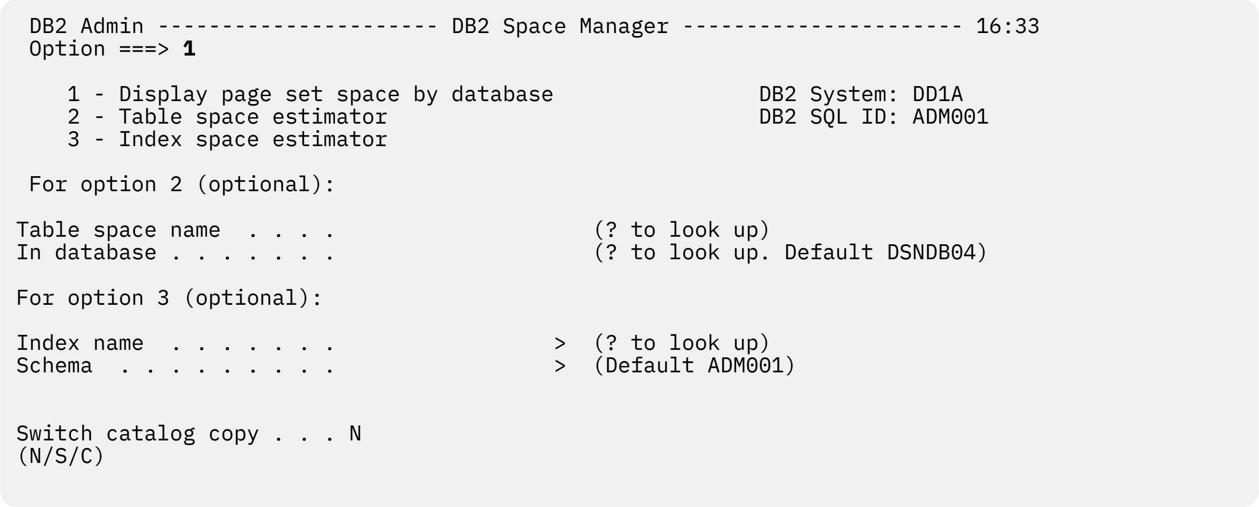
Displaying page set statistics

You can view page set statistics, such as the number of VSAM extents, the VSAM allocation type, the number of rows in the page set, and whether clustering is specified.

Procedure

To display page set statistics:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option SM, and press Enter.
2. On the **Space Manager (ADB2M)** panel, specify option 1, and press Enter.



```
 DB2 Admin ---------------------- DB2 Space Manager ---------------------- 16:33
 Option ===> 1

    1 - Display page set space by database                DB2 System: DD1A
    2 - Table space estimator                             DB2 SQL ID: ADM001
    3 - Index space estimator

 For option 2 (optional):

Table space name  . . . .                     (? to look up)
In database . . . . . . .                     (? to look up. Default DSNDB04)

For option 3 (optional):

Index name  . . . . . . .                  >  (? to look up)
Schema  . . . . . . . . .                  >  (Default ADM001)


Switch catalog copy . . . N
(N/S/C)
```

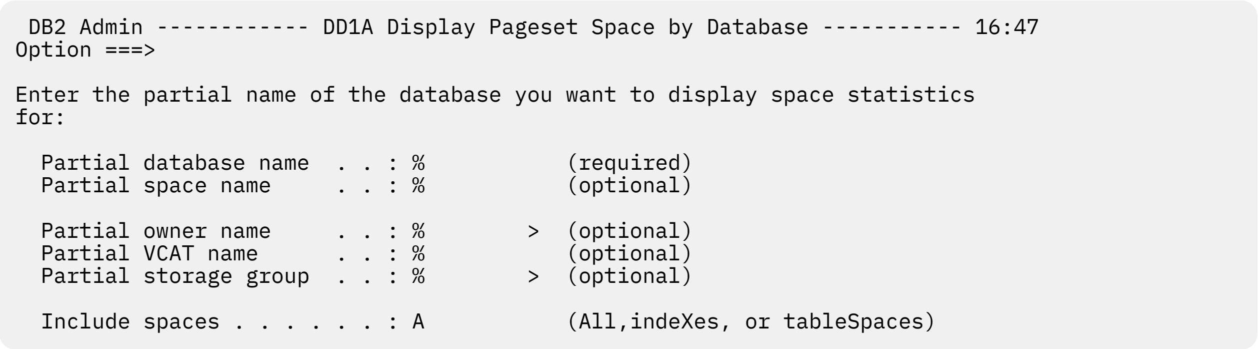
*Figure 347. **Space Manager (ADB2M)** panel*

3. On the **Space Management by Database (ADB2M1)** panel, specify the following information:
   - Specify a value for the **Partial database name** field. To improve performance, specify as much of the database name as possible.
   - Specify values for any of the optional fields. If you specify a value for the **Partial space name** field, specify as much of the space name as possible to improve performance.
   - Specify one of the following values for the **Include spaces** field:

     A
     Display both index and table space data

     X
     Display index data

     S
     Display table space data

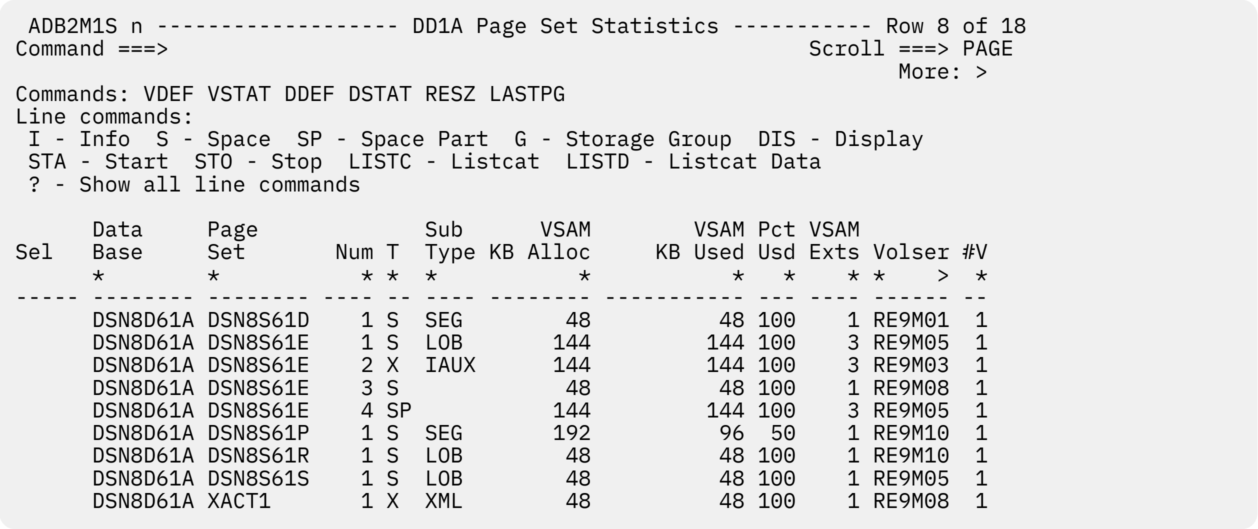


```
 DB2 Admin ------------ DD1A Display Pageset Space by Database ----------- 16:47
Option ===>

Enter the partial name of the database you want to display space statistics
for:

  Partial database name  . . : %           (required)
  Partial space name     . . : %           (optional)

  Partial owner name     . . : %        >  (optional)
  Partial VCAT name      . . : %           (optional)
  Partial storage group  . . : %        >  (optional)

  Include spaces . . . . . . : A           (All,indeXes, or tableSpaces)
```

Figure 348. ***Space Management by Database (ADB2M1)*** *panel*

4. Press Enter.

   The **Page Set Statistics (ADB2M1S)** panel displays VSAM-related page set data:



```
 ADB2M1S n ------------------- DD1A Page Set Statistics ----------- Row 8 of 18
Command ===>                                                  Scroll ===> PAGE
                                                                     More: >
Commands: VDEF VSTAT DDEF DSTAT RESZ LASTPG
Line commands:
 I - Info  S - Space  SP - Space Part  G - Storage Group  DIS - Display
 STA - Start  STO - Stop  LISTC - Listcat  LISTD - Listcat Data
 ? - Show all line commands

      Data     Page             Sub      VSAM        VSAM Pct VSAM
Sel   Base     Set       Num T  Type KB Alloc     KB Used Usd Exts Volser #V
      *        *           * *  *           *           *   *    * *    >  *
----- -------- -------- ---- -- ---- -------- ----------- --- ---- ------ --
      DSN8D61A DSN8S61D    1 S  SEG        48          48 100    1 RE9M01  1
      DSN8D61A DSN8S61E    1 S  LOB       144         144 100    3 RE9M05  1
      DSN8D61A DSN8S61E    2 X  IAUX      144         144 100    3 RE9M03  1
      DSN8D61A DSN8S61E    3 S              48          48 100    1 RE9M08  1
      DSN8D61A DSN8S61E    4 SP           144         144 100    3 RE9M05  1
      DSN8D61A DSN8S61P    1 S  SEG       192          96  50    1 RE9M10  1
      DSN8D61A DSN8S61R    1 S  LOB        48          48 100    1 RE9M10  1
      DSN8D61A DSN8S61S    1 S  LOB        48          48 100    1 RE9M05  1
      DSN8D61A XACT1       1 X  XML        48          48 100    1 RE9M08  1
```

Figure 349. ***Page Set Statistics (ADB2M1S)*** *panel for VSAM statistics*

To see additional statistics, scroll right. Alternatively, you can use the following commands to display only a particular set of statistics for the page data set:

VDEF
Displays VSAM definitions

VSTAT
Displays VSAM statistics. These statistics are listed on the first page when the panel is initially displayed.

DDEF
Displays Db2 definitions

DSTAT
Display Db2 statistics

LASTPG
Display the last logical page of statistics

Additionally, you can use the RESZ command to resize the page sets. For details, see "Resizing page sets" on page 633. You can also use the listed line commands to perform various space-related functions.

For details on any of these commands, see the panel help.

Resizing page sets

You can resize page sets to eliminate extents and to free unused space. To do so, use the Space Manager feature of Db2 Admin Tool. With Space Manager, you can resize all page sets for a database or select specific page sets to resize.

Procedure

To resize page sets:

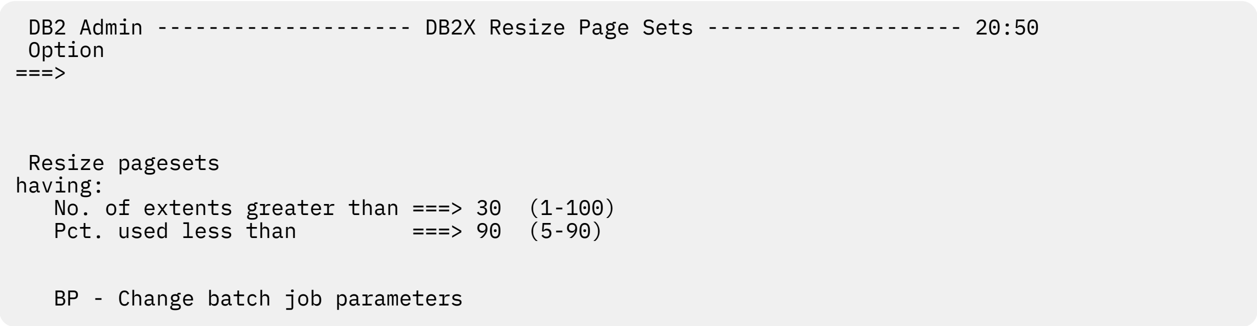
1. Complete steps 1 through 3 of "Displaying page set statistics" on page 631.
2. On the **Page Set Statistics (ADB2M1S)** panel, specify the RESZ line command to resize a specific page set or the RESZ primary command to resize all listed page sets, and press Enter.

   If the page set cannot be resized (because it is not overallocated or in extents), Db2 Admin Tool issues the following message:

   ```
   Nothing to resize (not overallocated or in extents)
   ```
3. On the **Resize Page Sets (ADB2M1R)** panel, complete the following fields, and press Enter:

   No. of extents greater than
   Specify the minimum number of extents that a page set must have to cause it to be resized.

   Pct. used less than
   Specify the percentage of space that must be available for a page space to be resized. For example, if you enter 45 in this field, only those page sets that are using less than 45 percent of the space available are resized.



```
 DB2 Admin -------------------- DB2X Resize Page Sets -------------------- 20:50
 Option
===>



 Resize pagesets
having:
   No. of extents greater than ===> 30  (1-100)
   Pct. used less than         ===> 90  (5-90)


   BP - Change batch job parameters
```

*Figure 350. **Resize Page Sets (ADB2M1R)** panel*

   Db2 Admin Toolcreates a batch job to resize those page sets that meet the criteria that you specified.

4. Submit the job to resize the page sets.

Moving between STOGROUP- and VCAT-related space

You can move a page set that is currently in a STOGROUP-defined space to a VCAT-defined space on another volume.

About this task

You can also move a page set that is currently in a VCAT-defined space to a STOGROUP-defined space.

If you enter the MOVE line command, you are prompted for additional input. The input asked for depends on whether you wish to move a STOGROUP-defined or a VCAT-defined page set.

To move between STOGROUP- and VCAT-related space, use the following procedure.

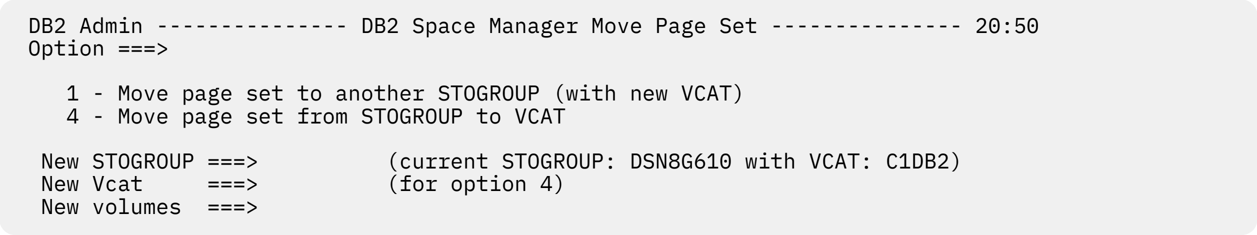
Procedure

1. Complete steps 1 through 3 of "Displaying page set statistics" on page 631.

   The **Page Set Statistics** for VSAM statistics panel is displayed.

2. Tab to the page set that you want to move and issue the MOVE line command.
3. In the panel that displays, enter additional information.

   If you are moving a STOGROUP-defined page set, the **Move Page Set Input** panel (ADB2M1M) is displayed, as shown in the following figure.



```
DB2 Admin --------------- DB2 Space Manager Move Page Set --------------- 20:50
Option ===>

   1 - Move page set to another STOGROUP (with new VCAT)
   4 - Move page set from STOGROUP to VCAT

 New STOGROUP ===>          (current STOGROUP: DSN8G610 with VCAT: C1DB2)
 New Vcat     ===>          (for option 4)
 New volumes  ===>
```

Figure 351. Move Page Set input panel (ADB2M1M): STOGROUP-defined page sets

1 - Move page set to another STOGROUP (with new VCAT)
If you select Option 1, Move page set to another STOGROUP (with new VCAT), you must enter the names of the new storage group, and optionally that of a new catalog.

New STOGROUP
Specify the name of the new storage group. The name of the current storage group and VSAM catalog are displayed for your information.

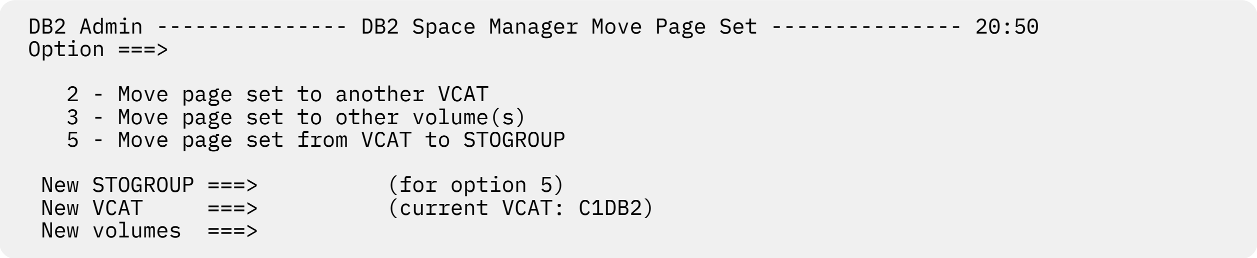
New Vcat
Specify the name of a VSAM catalog.

4 - Move page set from STOGROUP to VCAT
If you select Option 4, Move page set from STOGROUP to VCAT, you must enter the name of a new VSAM catalog, and optionally, the new volumes for the page set. Use commas to separate volume names.

New Vcat
Specify the name of a VSAM catalog.

New volumes
Optionally, specify the name of a new volume. For multiple volumes, separate the volume names with a comma.

If you are moving a VCAT-defined page set, the **Move Page Set Input** panel (ADB2M1M) is displayed, as shown in the following figure.



```
DB2 Admin --------------- DB2 Space Manager Move Page Set --------------- 20:50
Option ===>

   2 - Move page set to another VCAT
   3 - Move page set to other volume(s)
   5 - Move page set from VCAT to STOGROUP

 New STOGROUP ===>          (for option 5)
 New VCAT     ===>          (current VCAT: C1DB2)
 New volumes  ===>
```

Figure 352. Move Page Set input panel (ADB2M1M): VCAT-defined page set

2 - Move page set to another VCAT
If you select Option 2, Move page set to another VCAT, you must enter the name of the new VCAT, and optionally, the new volumes for the page set.

New VCAT
Specify the name of a VSAM catalog. The name of the current VCAT is displayed for your information.

New volumes
Specify the name of a new volume. For multiple volumes, separate the volume names with a comma.

3 - Move page set to other volume(s)

If you select Option 3, Move page set to other volume(s), enter the name(s) of one or more volumes.

New volumes

Specify the name of a new volume. For multiple volumes, separate the volume names with a comma.

5 - Move page set from VCAT to STOGROUP

If you select Option 5, Move page set from VCAT to STOGROUP, enter the name of a new STOGROUP.

New STOGROUP

Specify the name of the new storage group.

Estimating space requirements for table spaces

You can use the space manager feature of Db2 Admin Tool to estimate the space requirements for a table space. For example, if you plan to add a significant number of rows to a table, you might want to first determine how much space this addition requires.

About this task

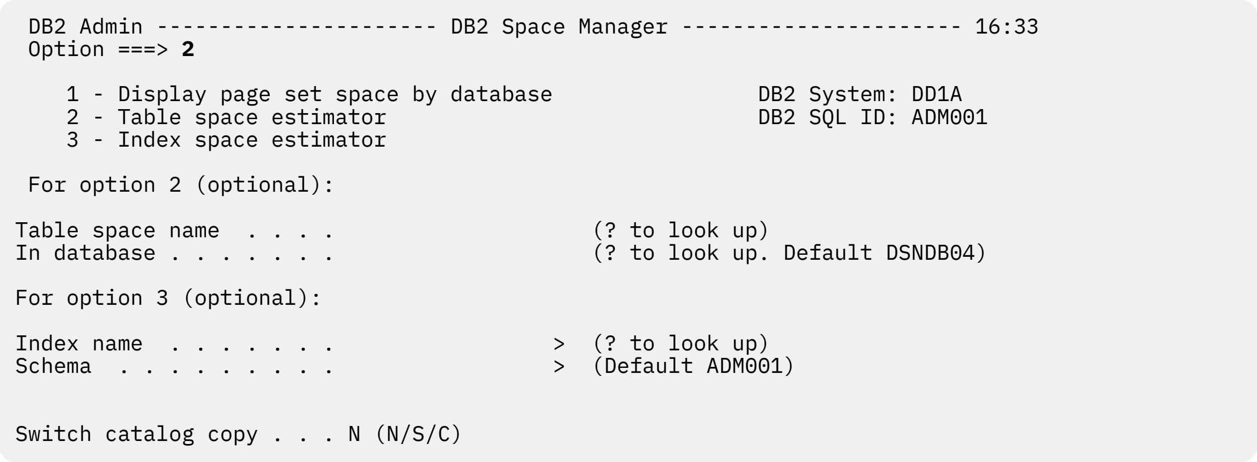
The space management features of estimating the space requirements for a table space and estimating the number of extents are available as REST APIs. See "Provided REST APIs" on page 876.

Procedure

To estimate the space requirements for a table space:

Tip: If you are viewing a table space on the **Table Spaces (ADB21S)** panel or a partition on the **Table Space Parts for table-space (ADB21SP)** panel, you can specify the SE line command to navigate directly to the space estimator and then start with step "3" on page 635.

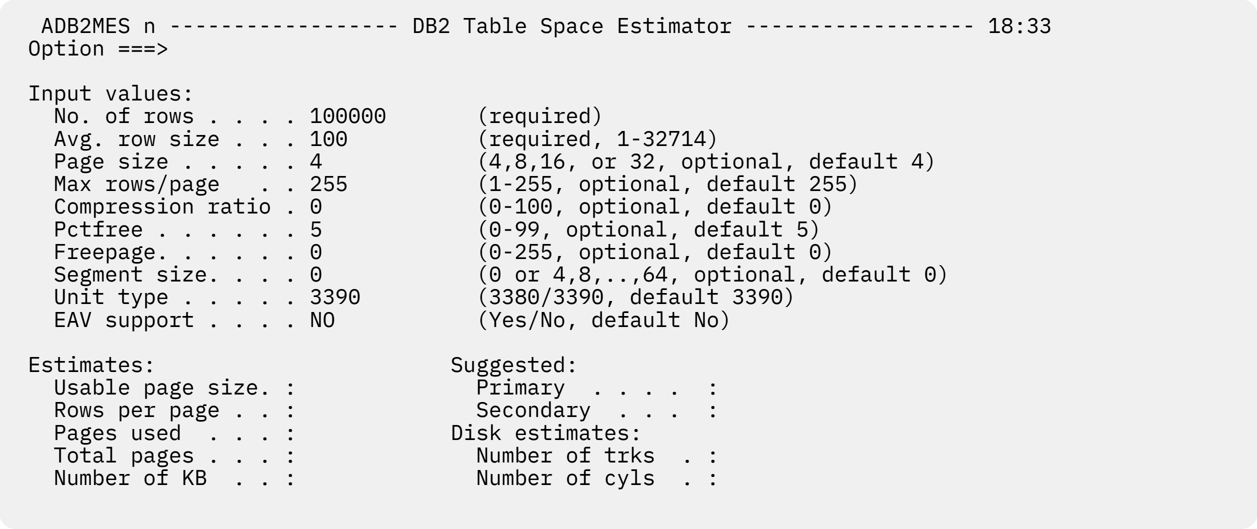
1. On the **DB2 Administration Menu (ADB2)** panel, specify option SM, and press Enter.
2. On the **Space Manager (ADB2M)** panel, specify option 2 and optionally a table space name, and press Enter:



```
 DB2 Admin ---------------------- DB2 Space Manager ---------------------- 16:33
 Option ===> 2

    1 - Display page set space by database               DB2 System: DD1A
    2 - Table space estimator                            DB2 SQL ID: ADM001
    3 - Index space estimator

 For option 2 (optional):

Table space name  . . . .                    (? to look up)
In database . . . . . . .                    (? to look up. Default DSNDB04)

For option 3 (optional):

Index name  . . . . . . .                 >  (? to look up)
Schema  . . . . . . . . .                 >  (Default ADM001)


Switch catalog copy . . . N (N/S/C)
```

*Figure 353. **Space Manager (ADB2M)** panel*

3. On the **DB2 Table Space Estimator (ADB2MES)** panel, specify information about the table space in the fields in the **Input values** section, and press Enter:



```
 ADB2MES n ------------------ DB2 Table Space Estimator ------------------ 18:33
Option ===>

Input values:
  No. of rows . . . . 100000       (required)
  Avg. row size . . . 100          (required, 1-32714)
  Page size . . . . . 4            (4,8,16, or 32, optional, default 4)
  Max rows/page   . . 255          (1-255, optional, default 255)
  Compression ratio . 0            (0-100, optional, default 0)
  Pctfree . . . . . . 5            (0-99, optional, default 5)
  Freepage. . . . . . 0            (0-255, optional, default 0)
  Segment size. . . . 0            (0 or 4,8,..,64, optional, default 0)
  Unit type . . . . . 3390         (3380/3390, default 3390)
  EAV support . . . . NO           (Yes/No, default No)

Estimates:                       Suggested:
  Usable page size. :              Primary  . . . .  :
  Rows per page . . :              Secondary  . . .  :
  Pages used  . . . :            Disk estimates:
  Total pages . . . :              Number of trks  . :
  Number of KB  . . :              Number of cyls  . :
```

Note: Initially, all fields on this panel are blank. Only the first two fields (**No. of rows** and **Avg. row size**) are required. For more information about each field, see the panel help.

*Figure 354. **DB2 Table Space Estimator (ADB2MES)** panel*

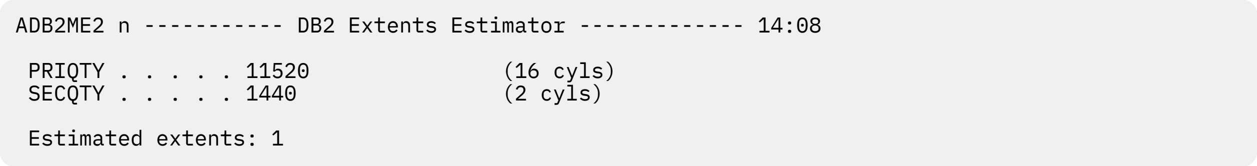
The value of the **Compression ratio** field is the percentage of rows that will not be compressed. For example, a compression value of 1 yields the maximum compression (because 99% of the rows are compressed). A compression value of 99 yields the minimum compression (because only 1% of the rows is compressed). A value of zero represents zero compression.

After you press Enter, the panel is refreshed. Based on the input values that you entered, the Table Space Estimator provides information about the estimated space that is required by the table space and suggests the amount of space to allocate for this table space. This information is displayed at the bottom of the panel. For more information about each field, see the panel help.

4. Optional: To see the estimated number of extents, specify the EXTENTS command and press Enter.

   Note: The EXTENTS command is available only after input values are specified, and you press Enter.

   On the resulting pop-up window you can override the **PRIQTY** and **SECQTY** field values (in KB) to see how the estimated number of extents changes:



```
ADB2ME2 n ----------- DB2 Extents Estimator ------------- 14:08

 PRIQTY . . . . . 11520              (16 cyls)
 SECQTY . . . . . 1440               (2 cyls)

 Estimated extents: 1
```

*Figure 355. **DB2 Extents Estimator (ADB2ME2)** panel*

Related information

How is space allocated on DASD volumes? (z/OS Basic Skills)

How Db2 extends data sets (Db2 12 for z/OS documentation)

Estimating space requirements for index spaces

You can use the space manager feature of Db2 Admin Tool to estimate the space requirements for an index space. For example, if you plan to add a significant amount of data to a table, you might want to determine how this additional will impact the space that is needed for an index.

About this task

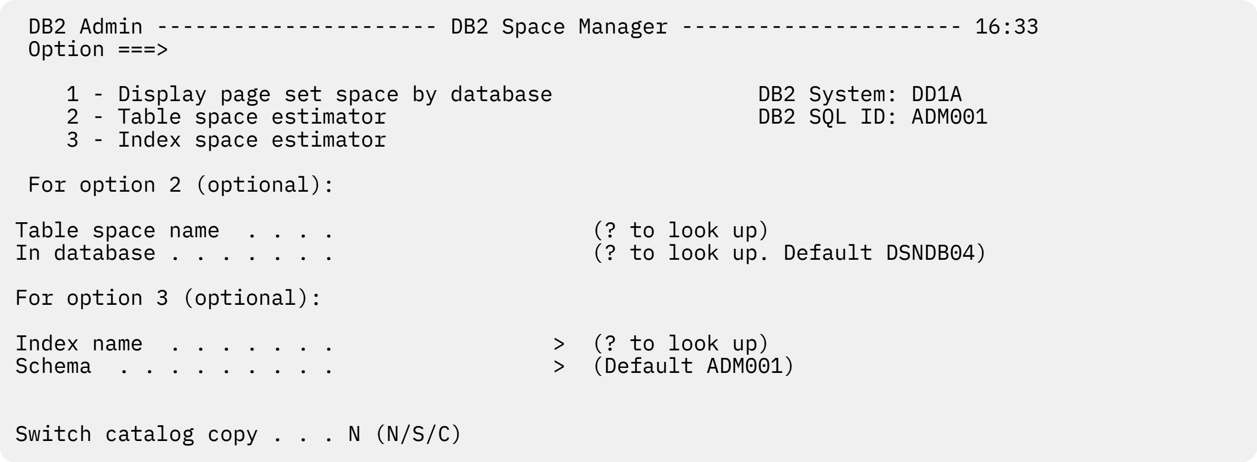
The space management features of estimating the space requirements for an index space and estimating the number of extents are available as REST APIs. See "Provided REST APIs" on page 876.

Procedure

To estimate the space requirements for an index space:

Tip: If you are viewing an index on the **Indexes (ADB21X)** panel or a partition on the **Index Parts (ADB21XP)** panel, you can specify the SE line command to navigate directly to the space estimator and then start with step "3" on page 637.

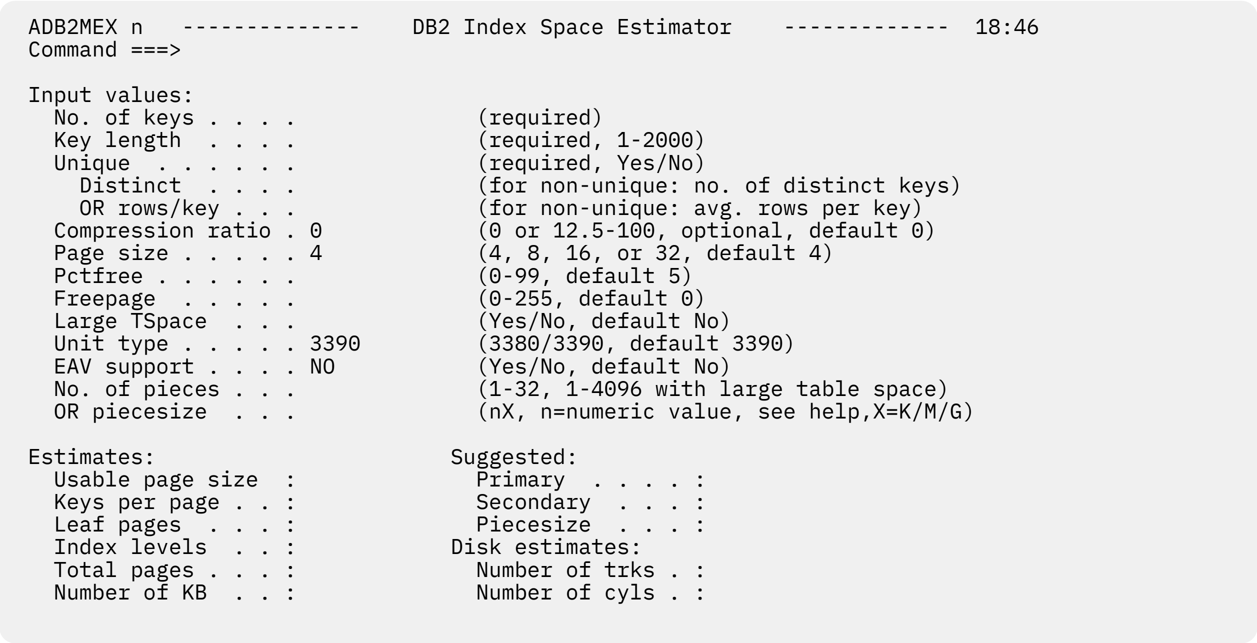
1. On the **DB2 Administration Menu (ADB2)** panel, specify option SM, and press Enter.
2. On the **Space Manager (ADB2M)** panel, specify option 3 and optionally an index name, and press Enter.



```
 DB2 Admin ---------------------- DB2 Space Manager ---------------------- 16:33
 Option ===>

    1 - Display page set space by database                DB2 System: DD1A
    2 - Table space estimator                             DB2 SQL ID: ADM001
    3 - Index space estimator

 For option 2 (optional):

Table space name  . . . .                    (? to look up)
In database . . . . . . .                    (? to look up. Default DSNDB04)

For option 3 (optional):

Index name  . . . . . . .                 >  (? to look up)
Schema  . . . . . . . . .                 >  (Default ADM001)


Switch catalog copy . . . N (N/S/C)
```

Figure 356. ***Space Manager (ADB2M)*** *panel*

3. On the **DB2 Index Space Estimator (ADB2MEX)** panel specify information about the index space in the fields in the **Input values** section, and press Enter:

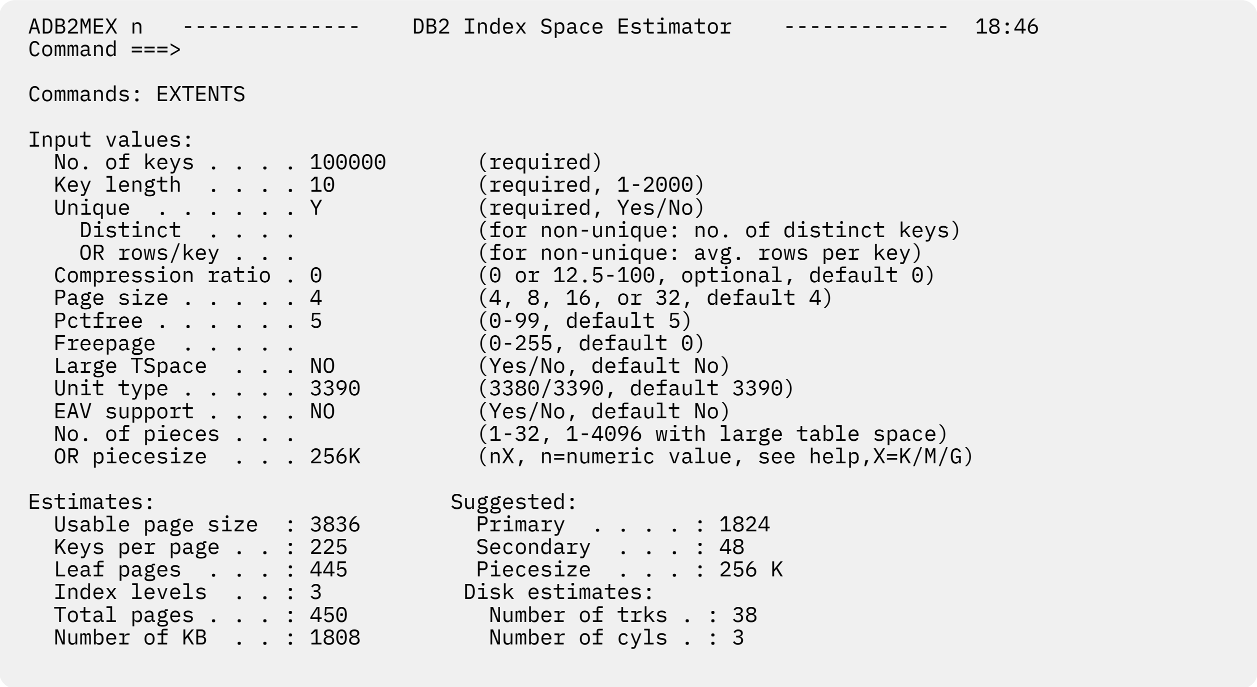


```
 ADB2MEX n   --------------    DB2 Index Space Estimator    -------------  18:46
 Command ===>

 Input values:
   No. of keys . . . .              (required)
   Key length  . . . .              (required, 1-2000)
   Unique  . . . . . .              (required, Yes/No)
     Distinct  . . . .              (for non-unique: no. of distinct keys)
     OR rows/key . . .              (for non-unique: avg. rows per key)
   Compression ratio . 0            (0 or 12.5-100, optional, default 0)
   Page size . . . . . 4            (4, 8, 16, or 32, default 4)
   Pctfree . . . . . .              (0-99, default 5)
   Freepage  . . . . .              (0-255, default 0)
   Large TSpace  . . .              (Yes/No, default No)
   Unit type . . . . . 3390         (3380/3390, default 3390)
   EAV support . . . . NO           (Yes/No, default No)
   No. of pieces . . .              (1-32, 1-4096 with large table space)
   OR piecesize  . . .              (nX, n=numeric value, see help,X=K/M/G)

 Estimates:                       Suggested:
   Usable page size  :              Primary  . . . . :
   Keys per page . . :              Secondary  . . . :
   Leaf pages  . . . :              Piecesize  . . . :
   Index levels  . . :            Disk estimates:
   Total pages . . . :              Number of trks . :
   Number of KB  . . :              Number of cyls . :
```

Note: Initially, all fields on this panel are blank. For more information about each field, see the panel help.

Figure 357. ***DB2 Index Space Estimator (ADB2MEX)*** *panel*

After you press Enter, the panel is refreshed. Based on the input values that you entered, the Index Space Estimator provides information about the estimated space that is required by the index space and suggests the amount of space to allocate for this index space. This information is displayed at the bottom of the panel:



```
ADB2MEX n   --------------    DB2 Index Space Estimator    -------------  18:46
Command ===>

Commands: EXTENTS

Input values:
  No. of keys . . . . 100000       (required)
  Key length  . . . . 10           (required, 1-2000)
  Unique  . . . . . . Y            (required, Yes/No)
    Distinct  . . . .              (for non-unique: no. of distinct keys)
    OR rows/key . . .              (for non-unique: avg. rows per key)
  Compression ratio . 0            (0 or 12.5-100, optional, default 0)
  Page size . . . . . 4            (4, 8, 16, or 32, default 4)
  Pctfree . . . . . . 5            (0-99, default 5)
  Freepage  . . . . .              (0-255, default 0)
  Large TSpace  . . . NO           (Yes/No, default No)
  Unit type . . . . . 3390         (3380/3390, default 3390)
  EAV support . . . . NO           (Yes/No, default No)
  No. of pieces . . .              (1-32, 1-4096 with large table space)
  OR piecesize  . . . 256K         (nX, n=numeric value, see help,X=K/M/G)

Estimates:                       Suggested:
  Usable page size  : 3836         Primary  . . . . : 1824
  Keys per page . . : 225          Secondary  . . . : 48
  Leaf pages  . . . : 445          Piecesize  . . . : 256 K
  Index levels  . . : 3           Disk estimates:
  Total pages . . . : 450           Number of trks . : 38
  Number of KB  . . : 1808          Number of cyls . : 3
```

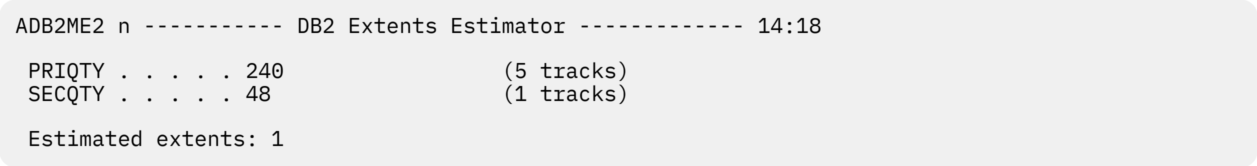
*Figure 358. **DB2 Index Space Estimator (ADB2MEX)** panel example*

For more information about each field, see the panel help.

4. Optional: To see the estimated number of extents, specify the EXTENTS command and press Enter.

   Note: The EXTENTS command is available only after input values are specified, and you press Enter.

   On the resulting pop-up window, you can override the **PRIQTY** and **SECQTY** field values (in KB) to see how the number of extents changes:



```
ADB2ME2 n ----------- DB2 Extents Estimator ------------- 14:18

 PRIQTY . . . . . 240                  (5 tracks)
 SECQTY . . . . . 48                   (1 tracks)

 Estimated extents: 1
```

*Figure 359. **DB2 Extents Estimator (ADB2ME2)** panel*

Related information

How is space allocated on DASD volumes? (z/OS Basic Skills)

How Db2 extends data sets (Db2 12 for z/OS documentation)

Change Management (CM)

The Change Management (CM) function in Db2 Admin Tool simplifies the process of recording and tracking the changes that you make to your Db2 objects. CM assigns a change ID for every change that you make to your Db2 objects and registers each change in the Change Management database.

CM provides the following features:

- Enables you to analyze how a change affects existing objects.
- Warns you if any pending changes exist on the same object that you plan to change. You can then specify whether your change should supersede or follow the pending changes.
- Allows you to define new changes as if any pending changes have already been made.

- Facilitates the generation of new versions to provide a snapshot of your database definitions after changes have been made.
- Provides an interface that allows you to track and query changes to objects and quickly find all of the components that are involved in a change.
- Provides an audit trail and helps automate the process of recovering changes.
- Maintains the relationships between changes, versions, masks, ignores, generated DDL, and unloaded data.
- Facilitates moving changes from one Db2 subsystem to another.

You can make changes by using the following features in Db2 Admin Tool or Db2 Object Comparison Tool:

- SQL CREATE, ALTER, DROP, RENAME, COMMENT, and LABEL statements that are executed from the input screen or from a data set
- SQL REVOKE statements that are executed from the input screen or from a data set as immediate changes
- The AL line command to change or rename a database
- The AL line command or ALT command to change a table space or index space
- The AL line command or ALT line command to change a table
- Comparisons in which changes are made to synchronize the target system with the source system
- Changes that are defined through the Change Management panels

Restriction: The GRANT USAGE ON JAR statement is not supported in CHANGE MANAGEMENT or in the Db2 Object Comparison Tool.

Change Management terminology

Before you use Change Management (CM), you should understand the terms that CM uses.

Change Management database
A database that contains several objects that are required by Db2 Admin Tool to manage changes.

exclude specification
A list of objects that you specify to be omitted from the compare process. The selected objects are not included as input to or output from the compare process.

fast change
A change that can or should be run immediately. If the affected objects have pending changes, the fast change is called an *emergency change*, and it supersedes the pending changes. If the affected objects do not have any pending changes, the fast change is called an *immediate change*.

ignore changes specification
A list of changes to objects from saved compare results that you specify to be ignored in subsequent compare processing. The selected object types participate in the compare process but changes to the object types are not propagated.

ignore fields
The Db2 catalog fields that are ignored when objects are compared.

masks
A specification of how names are to be translated when objects are compared or when they are moved from one system to another (source to target). Masks also allow you to overwrite the values of certain table space and index space attributes.

multi-target change
A change that is initially registered on one system (the "central" system) and can be used to distribute and track a change to database objects across one or more target systems. A separate change is registered and runs on each target system.

prerequisite change
A change that must be run before the current change is run. When you create a change for an object, the object might have *pending changes*, which are changes that have yet to be run. You can choose

to keep the pending changes as prerequisite changes for the new change or make your new change a *supersede change*, which puts the new change ahead of the pending changes.

recover change

A change that lets you back out a change that has been completed. Backing out a completed change requires determining whether the change has a recover change, whether other changes must be recovered first and in which order, and whether objects affected by the change have pending changes that have to be reanalyzed after the change is recovered. Db2 Admin Tool uses a *recover strategy* to determine all of this information for you.

version

A snapshot of a set of object definitions at a point in time. With Change Management, you have the option of creating a new base version before or after applying a change. You can then use this *base version* for a subsequent change or choose to generate DDL from the base version. In Change Management, the special type of version file that is called a *delta version* is no longer used.

version scope

The set of objects to be included in processing a version.

Prerequisites for Change Management

Using Change Management (CM) requires additional setup.

Before you can use Change Management (CM), the following requirements must be met:

- IBM Db2 Object Comparison Tool for z/OS must be installed.
- Db2 Admin Tool must be customized so that Change Management is enabled.
- If you want Change Management to be able take a snapshot of objects at a point in time (in a *version file*), ensure that the **Get DB2 ZPARM** option is set to YES on the **DB2 Admin Defaults (ADB2P2)** panel (GETDB2ZP='Y'). (For instructions on navigating to this panel and setting defaults, see "Changing defaults" on page 231.) This value ensures that GEN calls the Db2 stored procedure ADMIN_INFO_SYSPARM to get the Db2 subsystem parameter values. Change Management needs the Db2 subsystem parameter values (in the Db2 DSNZPARM initialization parameter module) to write the version file.
- Either the default Change Management level or the Change Management level for the current SQL ID must be set to either REQUIRED or OPTIONAL. To set one of these levels, use one of the following procedures:
  - "Setting the default Change Management level" on page 640
  - "Setting the Change Management level for specific SQL IDs" on page 641

Setting the default Change Management level

The *Change Management level* determines whether changes must be registered, can be registered, or cannot be registered in the Change Management database. The default value determines the level if a one is not specifically defined for the current SQL ID.

About this task

The Change Management levels are:

REQUIRED

All changes must be registered in the Change Management database.

OPTIONAL

Changes can be registered in the Change Management database. When you define a change, you are prompted whether to make the change through Change Management.

Tip: If you have a set of objects that are being managed under Change Management, all changes to those objects should be made through Change Management.

You might set the Change Management level to OPTIONAL in the following situations:

- When you are testing Change Management
- When you have a set of objects that are being managed under Change Management, and you can ensure that the SQL ID that makes changes to those objects will choose Change Management when prompted.

NONE
No changes can be registered in the Change Management database.

The default Change Management level is stored in the Change Management ID table.

Procedure

To set the default Change Management level:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option 5, and press Enter.
3. On the **Manage ID Table (ADB2C5)** panel, specify REQUIRED, OPTIONAL, or NONE in the **Change management default level setting** field.
4. Issue the SAVE command to update the ID table.

Related concepts

"Change Management terminology" on page 639
Before you use Change Management (CM), you should understand the terms that CM uses.

Related tasks

"Setting the Change Management level for specific SQL IDs" on page 641
You can override the default Change Management level for one or more SQL IDs. The level that you define for each SQL ID is stored as an entry in the Change Management ID table.

Setting the Change Management level for specific SQL IDs

You can override the default Change Management level for one or more SQL IDs. The level that you define for each SQL ID is stored as an entry in the Change Management ID table.

About this task

For information about the default Change Management level and possible values, see "Setting the default Change Management level" on page 640.

Procedure

To specify the Change Management level for a specific SQL ID:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option 5, and press Enter.
3. On the **Manage ID Table (ADB2C5)** panel, perform one or more of the following actions:

   - To add a new SQL ID, issue the I line command, and specify the SQL ID and the Change Management level for that ID.
   - To change the Change Management level for an existing SQL ID, type over the current value in the **Level** column.

   Possible values for the **Level** column are REQUIRED, OPTIONAL, or NONE.
4. Issue the SAVE command to update the ID table.

Related concepts

"Change Management terminology" on page 639

Before you use Change Management (CM), you should understand the terms that CM uses.

Recommendations for designing a Change Management strategy

An effective change management strategy is one that is well planned. The most important factor to consider is to ensure that changes to a set of objects are either all performed through Change Management (CM) or are all performed without Change Management.

Requiring that all changes go through Change Management is easy when the objects that should go through Change Management are handled by a few SQL IDs and the SQL IDs are used only for these objects. If the SQL IDs are also being used to change objects that should not go through Change Management, you should set the Change Management level option to OPTIONAL, and the user will have to decide whether the change should go though Change Management.

A few example Change Management strategies are:

- When Change Management is being used for the objects for only one application:
  - Set the Change Management level for the SQL ID that is used to manage the objects for the application to REQUIRED.
  - Set the level for the other SQL IDs to NONE by setting the default Change Management level to NONE.
- When Change Management is being used for the objects for all applications except for a few objects that are under design and development:
  - Set the default Change Management level to REQUIRED.
  - Set the Change Management level for the SQL IDs that are used to change the objects that are under design and development to NONE. If those SQL IDs are also used to change objects that are not under design and development, set the Change Management level for the SQL IDs to OPTIONAL; the user will need to specify whether to use Change Management upon each change.
- When Change Management is being tested:
  - Set the default Change Management level to OPTIONAL.

Change Management process

The basic process of making a change by using Change Management (CM) has four steps: define the change, register the change, analyze the change, and run the change.

1. Define a change

Changes can come from a variety of sources. For example, you can use the regular features of Db2 Admin Tool or Db2 Object Comparison Tool to generate Db2 object changes, import statements from a data set into a change, or import changes that have been promoted from other Db2 subsystems.

Handling pending changes

The objects that you are changing might have pending changes, which are changes that are managed by Change Management and are not yet complete. You need to specify whether the change that you are defining should supersede these pending changes.

When and where you specify how pending changes are to be handled depends on whether the pending changes can be applied as virtual changes. Pending changes can be treated as virtual changes when you make changes by using one of the following methods:

- The ALTER dialogs (the ALT function) to rename a database or redefine a table space, index, or table.
- The CREATE dialogs (option 2.4 from the Db2 Admin Tool main menu) to create a table space, table, index, materialized table, view, or trigger.
- The **Tables, Views, and Aliases (ADB21T)** panel to rename a table

When the pending changes can be applied as virtual changes, you specify how to handle the pending changes at the time you define the change. When you define the change, the list of pending changes is displayed. You must specify whether to apply these pending changes and define your new change based on a virtual representation of the objects with the pending changes applied. When you choose to supersede the pending changes, you define your change without considering the effect of the pending changes, and the new change becomes a prerequisite change for the pending changes.

Important: For performance reasons, to minimize the amount of time spent traversing relationships, especially for renames, the list of pending changes that Db2 Admin Tool displays might not be complete. However, if you apply the pending changes, all of pending changes for the objects are applied regardless of whether they are displayed in the list.

Tip: To minimize the amount of time that it takes to apply pending changes, keep the number of uncompleted changes (those changes in the DEFINED, ANALYZED, or RUNNING status) to a minimum.

When the pending changes cannot be applied as virtual changes, you are prompted to specify how the pending changes are to be handled at the time you register the change (or shortly before you register the change when the change is being imported from a data set or from other systems or when performing a comparison in Db2 Object Comparison Tool). You must determine whether your new change is to be added to a pending change, completed before or after any existing pending changes, or executed immediately.

Exception: You are not prompted to specify how to handle pending changes if you use the Change Management panels to define a change. In this case, you insert a change on the panel and then create change statements for the change.

2. Register a change

After you define a change, Db2 Admin Tool prompts you to register the change in the Change Management database. You specify a name for the change, and Db2 Admin Tool automatically assigns a change ID to the change.

Depending on the method that was used to define the change, you might be prompted to specify whether to register the change as a normal change, a multi-target change, an emergency change, or an immediate change and how to handle pending changes for the objects that are involved in the change:

- If pending changes exist, you can register the change as a normal change or an emergency change. If you register the change as a normal change, you also must specify whether the change should be made before or after the pending changes.
- If no pending changes exist, you can register the change as a normal change or an immediate change.

Db2 Admin Tool runs emergency and immediate changes immediately. The analyze and run phases do not apply.

3. Analyze a change

A normal change must be analyzed before the change can be applied to the objects. When you issue the command to analyze a change, Db2 Admin Tool generates a batch job that you submit. The batch job analyzes how the change modifies existing objects, both in Db2 and in any of the prerequisite changes, and creates a work statement list (WSL) that will be used to run the changes. During the analyze process, the embedded SQL statements semantics are checked and Db2 Admin Tool automatically generates two new base versions:

- A target version, which represents the objects in the Db2 catalog plus any prerequisite changes
- A source version, which is the target version plus the changes for the change that is being analyzed

Db2 Admin Tool then invokes Db2 Object Comparison Tool to compare the source and target base versions to generate a WSL that will be used in the run process to apply the changes. The base versions that are used in this process are temporary and are not saved. For more information about the methods Db2 Admin Tool uses to generate the base version, see "Base version method for change analysis" on page 655

When you analyze a change, you can specify that a *recover change* be created. Creating a recover change gives you the option of backing out the change. When you first choose to create a recover change, you are prompted to register the recover change. The recover change is automatically updated if the original change is reanalyzed.

4. Run a change

After a change has been successfully analyzed, it is ready to be run. That is, you are ready to apply the change to the database. When you issue the command to run the change, Db2 Admin Tool creates a batch job that you submit. The batch job runs the WSL that was generated during the analyze process.

If the change has prerequisite changes, you cannot run the change, and Db2 Admin Tool will prompt you to run the prerequisite changes first.

The run job performs a task called *runtime analyze*, which ensures that the Db2 catalog has not changed from the time the change was analyzed. The run job reanalyzes the change and creates a second WSL, using the current Db2 catalog and the automatic base version (A) method. The second WSL is compared with the WSL that was generated during the normal analyze. If the DDL and DCL content are the same, the run job continues and the WSL that was generated during the normal analyze is run to apply the changes. If the DDL or DCL content are different, the run job stops with a return code of 8, and the change is not run.

When you run a change, you have the option of specifying that a new base version be created after the changes have been made successfully. If you want to have a new base version created, a version scope that defines the set of objects to be included in the base version must exist.

Important: Always use the RN command on the **CM - Changes (ADB2C11)** panel to run changes that are being managed under Change Management. Do not run the WSL that was generated during the analyze process directly from the **Work Statement List Library (ADB2W1)** panel, because Db2 Admin Tool cannot track changes that are made outside of the Change Management process. Also, do not use the line commands on the **Work Statement List Library (ADB2W1)** panel to edit, delete, copy, append, or clone a WSL that was generated during the analyze process.

Additional tasks

Although the basic process of making a change has four steps (define, register, analyze, and run), you can also use Change Management to complete additional tasks, such as:

- Recovering changes
- Tracking changes and changed objects
- Managing masks, ignores, versions, and version scopes
- Promoting changes from one system to another

Related concepts

“Change Management terminology” on page 639
Before you use Change Management (CM), you should understand the terms that CM uses.

Types of changes and change status

To facilitate change management, Db2 Admin Tool categorizes changes into several types and assigns a status to each change as it moves through the change management process.

The type is assigned when a change is registered. The following table describes the types of changes:

Table 33. Types of changes

Type of change	Description
CHANGE	A change that is defined through the usual change functions in Db2 Admin Tool and Db2 Object Comparison Tool (such as ALT and compare) and then goes through the Change Management process of being analyzed and run.
MULTI-TC	A change that is generated on a central system for the purpose of importing it to multiple target systems. On target systems, the changes that are registered are then analyzed and run to apply the changes to the target catalog.

Table 33. Types of changes (continued)

Type of change	Description
FAST	A change that is run immediately. If no pending changes exist, the fast change is called an *immediate change*. If pending changes exist for the object or related objects that are affected by the fast change, the fast change is called an *emergency change*, and it supersedes the pending changes. The pending changes are placed in DEFINED status. Because fast changes are run immediately upon registration, you cannot analyze or run them manually. You also cannot modify fast changes, recover them, or promote them to other systems.
COMPARE	A change that is generated by comparing two items, such as two base versions, two DDL files, two catalog objects, or a DDL file and a catalog object.
PROMOTE	A change that is generated by importing statements from a data set or a changes file.
RECOVER	A change that was automatically generated to back out another change. When you analyze a change, you have the option of having a recover change created. Db2 Admin Tool generates a recover change, assigns a change ID to the recover change, and puts the recover change in ANALYZED status. To recover a change, issue the RC line command for the original change. Do not issue the RN line command for the recover change.

The status of a change is updated when actions are taken on the change. The following table describes the possible values for the status:

Table 34. Status of changes

Status	Description
INITIAL	The change was created, but its registration in the Change Management database is incomplete. You can try to register the change by issuing the restart line command (RST) on the **CM - Changes (ADB2C11)** panel. In this case however, Db2 Admin Tool cannot detect and process any prerequisite changes that might exist. You must identify any prerequisite changes yourself and reanalyze any change in ANALYZED status to ensure its validity.
DEFINED	The change was created and registered in the Change Management database. The change is ready to be analyzed.
ANALYZED	The change was validated and a WSL to run the change was generated. The change is ready to be run.
RUNNING	The change is currently running. A RUNNING status that does not change to COMPLETE status indicates that the job to run the change failed at some point.
COMPLETE	The change ran successfully.
CANCELED	The change was canceled.
FAILED	The change is a fast change that was run immediately but did not complete successfully.

Related concepts

"Change Management terminology" on page 639

Before you use Change Management (CM), you should understand the terms that CM uses.

Making changes by using Change Management (CM)

Using Change Management (CM) to make changes to your Db2 objects can help simplify the process of tracking and managing changes.

Before you begin

Complete all actions in "Prerequisites for Change Management" on page 640.

About this task

Restriction: The value of the character input fields on the Change Management panels cannot contain an apostrophe (or single quotation mark). For example, do not specify an apostrophe in the name of any change, version, mask, or ignore.

Procedure

To make a change by using Change Management:

1. Define the change.

   For example, performing one of the following tasks can create a change:

   - "Altering a table by using the ALT line command" on page 461
   - "Running SQL statements from a data set" on page 319
   - "Running SQL statements from screen input" on page 318

   If Change Management is optional for your SQL ID, the specify YES when you are prompted whether to use Change Management on the **Change Management Prompt (ADB2CMPR)** panel.

   If the **Pending Changes - Conflict Resolution** panel is displayed with a list of pending changes for the affected objects, specify whether to apply the pending changes as virtual changes before you continue to define your change.
2. Register the change by completing the steps in "Registering a change" on page 648.
3. Analyze the change.

   To analyze the change by using panels, complete the steps in "Analyzing a change" on page 652.

   To analyze the change by using the Change Management batch interface, see "Managing Changes by using the CM batch interface" on page 659.
4. Run the change.

   To run the change by using panels, complete the steps in "Running a change" on page 656.

   To run the change by using the Change Management batch interface, see "Managing Changes by using the CM batch interface" on page 659.

Related concepts

"Change Management process" on page 642
The basic process of making a change by using Change Management (CM) has four steps: define the change, register the change, analyze the change, and run the change.

"Change Management (CM)" on page 638
The Change Management (CM) function in Db2 Admin Tool simplifies the process of recording and tracking the changes that you make to your Db2 objects. CM assigns a change ID for every change that you make to your Db2 objects and registers each change in the Change Management database.

"Change Management terminology" on page 639

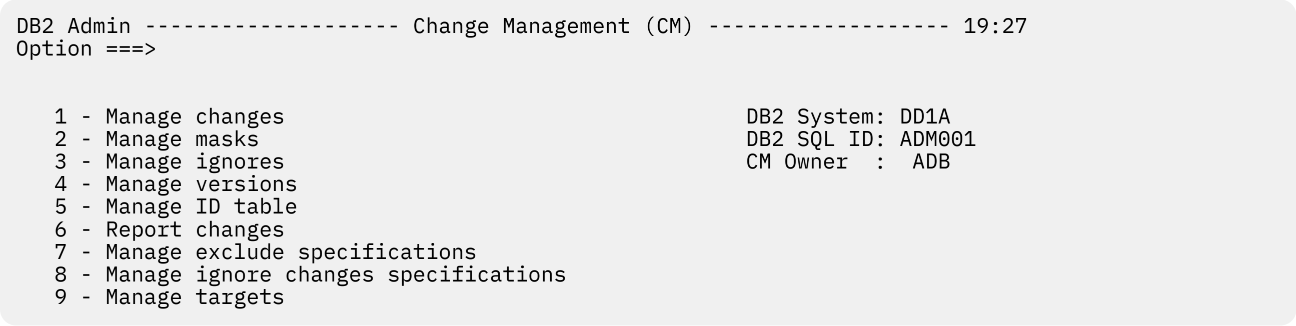
Before you use Change Management (CM), you should understand the terms that CM uses.

The Change Management main menu panel

The **Change Management (CM) (ADB2C)** panel is the main menu for accessing Change Management functions.

To display the **Change Management (CM) (ADB2C)** panel, specify the CM option on the **DB2 Administration Menu (ADB2)** panel.

Tip: You can also issue the CMM special command from any Db2 Admin Tool panel to go directly to the **Change Management (CM) (ADB2C)** panel.



```
DB2 Admin -------------------- Change Management (CM) ------------------- 19:27
Option ===>


   1 - Manage changes                                    DB2 System: DD1A
   2 - Manage masks                                      DB2 SQL ID: ADM001
   3 - Manage ignores                                    CM Owner  :  ADB
   4 - Manage versions
   5 - Manage ID table
   6 - Report changes
   7 - Manage exclude specifications
   8 - Manage ignore changes specifications
   9 - Manage targets
```

*Figure 360. **Change Management (CM) (ADB2C)** panel*

The following options are available on this panel:

1 - Manage changes
Select this option to display changes (and subsequently analyze, run, or recover a change), create a new change, promote a change, or import a change.

2 - Manage masks
Select this option to display the masks that are defined or create a new mask.

3 - Manage ignores
Select this option to display the ignores that are defined or create a new ignore.

4 - Manage versions
Select this option to display versions and version scopes or create a version scope.

5 - Manage ID table
Select this option to change the default Change Management level or to override the default level for specific SQL IDs.

6 - Report changes
Select this option to display changes or changed objects.

7 - Manage exclude specifications
Select this option to create, edit or display exclude specifications.

8 - Manage ignore changes specifications
Select this option to display ignore changes.

9 - Manage targets
Select this option to display or create targets for a change.

Registering a change

When you define a change and Change Management is required (or Change Management is optional and you have specified to use Change Management), Db2 Admin Tool automatically prompts you to register the change in the Change Management database.

Before you begin

This procedure assumes that you have already defined a change, as described in "Making changes by using Change Management (CM)" on page 646 and the **Register Change (ADB2CMRG)** panel or the **Register Options (ADB2CRO)** panel is displayed.

About this task

For more information about registering a change, see "2. Register a change" on page 643.

Requirement: If you choose to use Change Management after selecting option 1A (runs all statements) on the **Statement Execution Prompt (ADB2PSTM)** panel, you still must register each statement individually. You need to navigate through the **Change Management Prompt (ADB2CMPR)** panel and the **Register Change (ADB2CMRG)** panel for each statement. However, you can make each statement part of the same change by specifying the same change owner and name combination for each statement.

Procedure

To register a change:

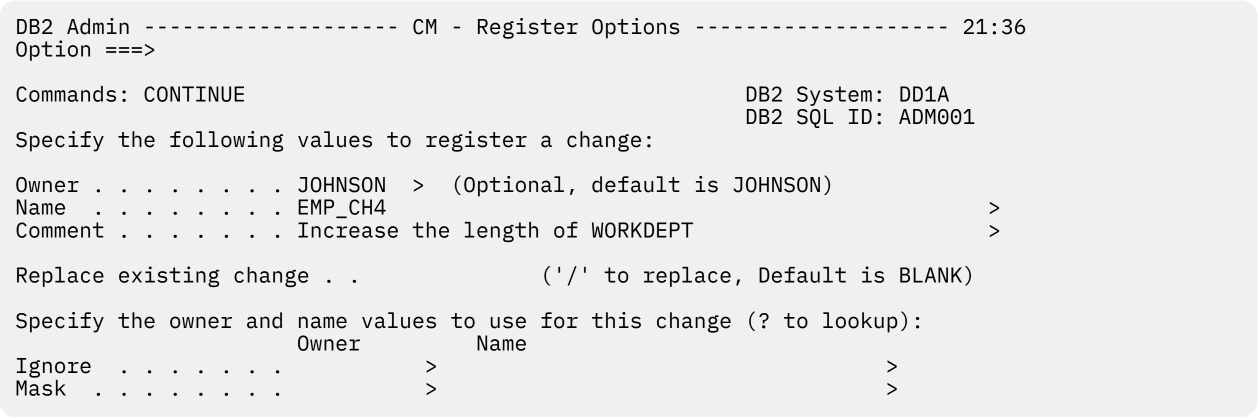
1. If the **Register Change (ADB2CMRG)** panel is displayed, specify the type of change and press Enter. (Whether this panel is displayed depends on how the change was defined.)

   If you register the change as an immediate or emergency change, specify an owner and name for the change, and optionally, a comment. When you press Enter, the change runs immediately and you can skip the remaining steps.

   If you register the change as a normal change and press Enter, the **Register Options (ADB2CRO)** panel is displayed.
2. On the **Register Options (ADB2CRO)** panel, complete the following information and issue the CONTINUE command:

   - Specify an owner and a name for the change. The default owner is the current SQL ID. If you specify the name of a existing change, the change statements are included in the existing change, if possible.

     You can include the change statements in an existing change when the existing change has no prerequisite changes and the existing change is not a recover change, a fast change, or a promote change on the source side.
   - Optionally, specify a comment for the change.
   - Specify whether you want to replace an existing change. If you leave the field blank no existing changes are replaced.



```
DB2 Admin -------------------- CM - Register Options -------------------- 21:36
Option ===>

Commands: CONTINUE                                       DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
Specify the following values to register a change:

Owner . . . . . . . . JOHNSON  >  (Optional, default is JOHNSON)
Name  . . . . . . . . EMP_CH4                                               >
Comment . . . . . . . Increase the length of WORKDEPT                       >

Replace existing change . .              ('/' to replace, Default is BLANK)

Specify the owner and name values to use for this change (? to lookup):
                      Owner         Name
Ignore  . . . . . . .          >                                   >
Mask  . . . . . . . .          >                                   >
```

Figure 361. ***Register Options (ADB2CRO)*** *panel*

If the first statement of the change is not a SET SCHEMA statement and if the value of CURRENT SCHEMA is different from the value of CURRENT SQLID, Db2 Admin Tool inserts a SET SCHEMA statement into the change table (ADBCHG) before processing other statements. Also, the LASTSCHEMA column of ADBCHG is updated with the current SCHEMA.

When more statements are added to an existing change, Db2 Admin Tool checks the LASTSCHEMA column of ADBCHG against the current schema and, if they are different, inserts another SET SCHEMA statement.

When the register process is triggered by **Restart** or **Editing the change statements via CM panels**, SET SCHEMA statements are not inserted. However, the LASTSCHEMA column in the ASDBCHG table is updated.

3. Optional: Verify that the change was registered and is in DEFINED status by completing the following steps:

   a) Issue the CMM command, and press Enter.

   b) On the **Change Management (CM) (ADB2C)** panel, specify option 1.

   c) On the **Manage Changes (ADB2C1)** panel, specify option 1.

   d) On the **Changes (ADB2C11)** panel, verify that your change is included in the list of changes.

   If the change is not registered successfully and is placed in INITIAL status, you can issue the restart line command (RST) to attempt to complete registration. However, when you restart the change, Db2 Admin Tool cannot detect and process any pending changes that might exist. You must identify any pending changes yourself and reanalyze any change in ANALYZED status to ensure its validity.

Results

Db2 Admin Tool has registered your change in the Change Management database and has automatically assigned a change ID to it.

What to do next

“Analyzing a change” on page 652

Related concepts

“Change Management terminology” on page 639
Before you use Change Management (CM), you should understand the terms that CM uses.

Related tasks

“Example: Registering a change that is defined with the ALT command” on page 650
This example shows how to register a change when pending changes can be applied as virtual changes before you define your change, such as when you use the ALT command to redefine a table.

“Example: Registering a change that is created from screen input” on page 651
This example shows how to register a change when you are prompted on the **Register Change (ADB2CMRG)** panel to specify whether to register the change as a normal or a fast change (emergency

or immediate) and how any pending changes are to be resolved. This example assumes that the affected objects have pending changes.

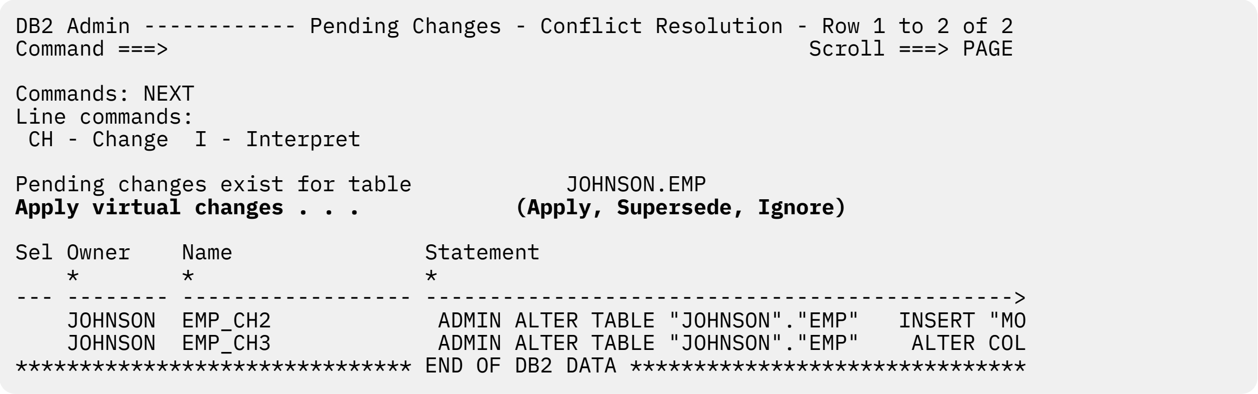
Example: Registering a change that is defined with the ALT command

This example shows how to register a change when pending changes can be applied as virtual changes before you define your change, such as when you use the ALT command to redefine a table.

Procedure

1. Issue the ALT command for the table that you want to change.
2. If Change Management is optional for your SQL ID, specify YES when you are prompted whether to use Change Management on the **Change Management Prompt (ADB2CMPR)** panel.
3. If Db2 Admin Tool displays a list of pending changes that are registered in Change Management for the affected object, specify how the pending changes are to be handled, and issue the NEXT command.

   The following figure shows an example of an object that has pending changes:



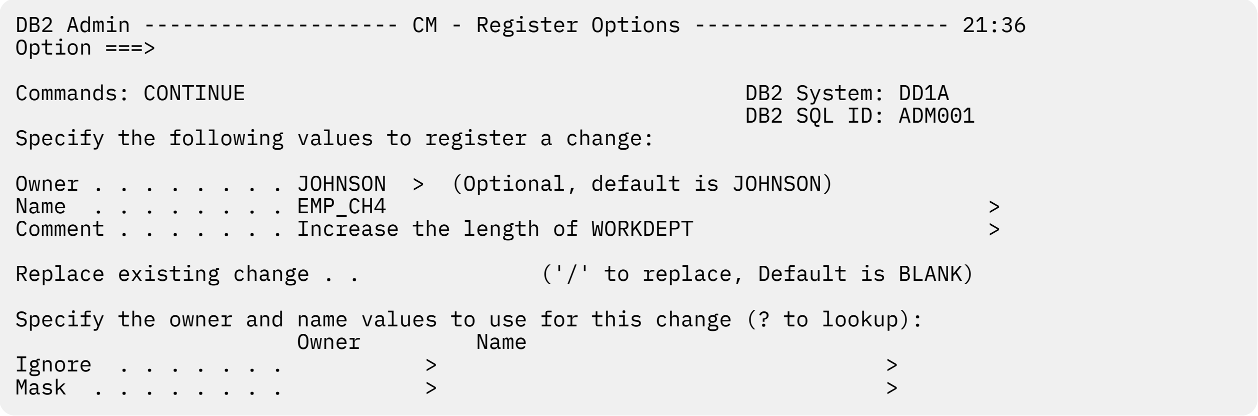
```
DB2 Admin ------------ Pending Changes - Conflict Resolution - Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

Commands: NEXT
Line commands:
 CH - Change  I - Interpret

Pending changes exist for table            JOHNSON.EMP
Apply virtual changes . . .            (Apply, Supersede, Ignore)

Sel Owner    Name               Statement
    *        *                  *
--- -------- ------------------ -------------------------------------------->
    JOHNSON  EMP_CH2             ADMIN ALTER TABLE "JOHNSON"."EMP"   INSERT "MO
    JOHNSON  EMP_CH3             ADMIN ALTER TABLE "JOHNSON"."EMP"    ALTER COL
******************************* END OF DB2 DATA *******************************
```

   Figure 362. Pending Changes - Conflict Resolution panel (ADB2CCR)

   When you apply the pending changes, you define your new changes based on a virtual representation of the objects as if the pending changes were made.

   When you supersede the pending changes, you define the new changes without taking into account the effect of any pending changes; the new change becomes a prerequisite change for the pending changes, and any pending changes that are in ANALYZED status are set to DEFINED status.

   When you ignore the pending changes, the new change you define does not become a prerequisite change for the pending changes. Any pending changes that are in ANALYZED status are not set to DEFINED status.
4. Complete the **Name** field and any other options that you want to specify on the **Register Options (ADB2CRO)** panel, and issue the CONTINUE command.



```
DB2 Admin -------------------- CM - Register Options -------------------- 21:36
Option ===>

Commands: CONTINUE                                     DB2 System: DD1A
                                                       DB2 SQL ID: ADM001
Specify the following values to register a change:

Owner . . . . . . . . JOHNSON  >  (Optional, default is JOHNSON)
Name  . . . . . . . . EMP_CH4                                          >
Comment . . . . . . . Increase the length of WORKDEPT                  >

Replace existing change . .             ('/' to replace, Default is BLANK)

Specify the owner and name values to use for this change (? to lookup):
                      Owner         Name
Ignore  . . . . . . .          >                                  >
Mask  . . . . . . . .          >                                  >
```

   Figure 363. Register Options panel (ADB2CRO)

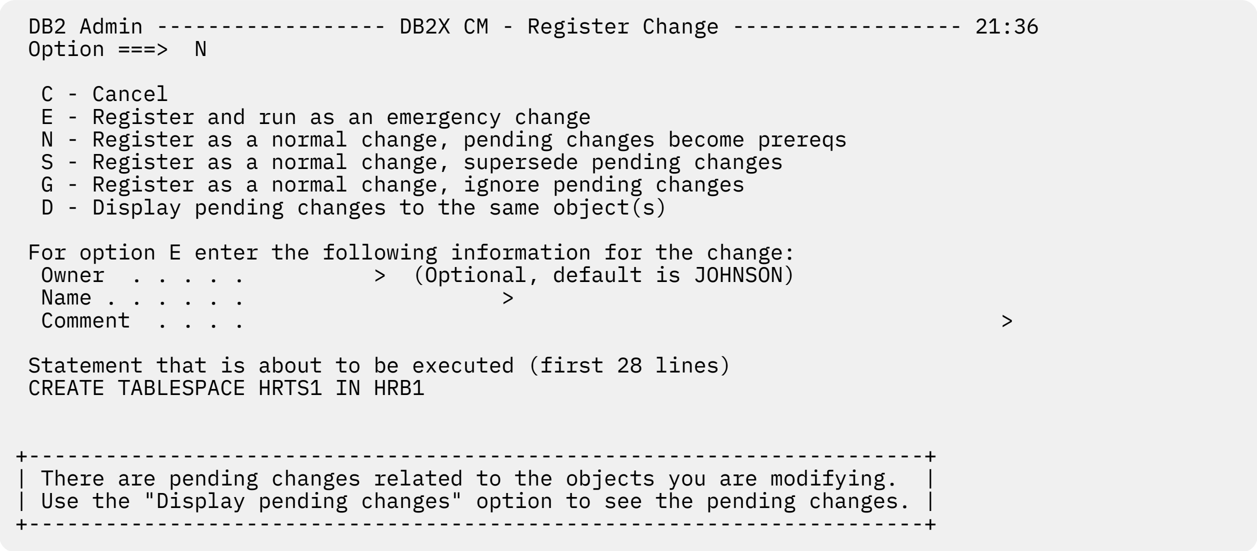
Note: When you make changes through ALT, and choose to apply virtual changes, the **Replace existing change** field is not editable.

Example: Registering a change that is created from screen input

This example shows how to register a change when you are prompted on the **Register Change (ADB2CMRG)** panel to specify whether to register the change as a normal or a fast change (emergency or immediate) and how any pending changes are to be resolved. This example assumes that the affected objects have pending changes.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2.1 to execute SQL statements from screen input.
2. Enter the SQL statements that you want to run and press Enter.
3. Press PF3 to exit.
4. If Change Management is optional for your SQL ID, specify YES when you are prompted whether to use Change Management on the **Change Management Prompt (ADB2CMPR)** panel.
5. On the **Register Change (ADB2CMRG)** panel, specify how the change should be registered, and press Enter:



```
 DB2 Admin ------------------ DB2X CM - Register Change ------------------ 21:36
 Option ===>  N

  C - Cancel
  E - Register and run as an emergency change
  N - Register as a normal change, pending changes become prereqs
  S - Register as a normal change, supersede pending changes
  G - Register as a normal change, ignore pending changes
  D - Display pending changes to the same object(s)

 For option E enter the following information for the change:
  Owner  . . . . .          >  (Optional, default is JOHNSON)
  Name . . . . . .                    >
  Comment  . . . .                                                       >

 Statement that is about to be executed (first 28 lines)
 CREATE TABLESPACE HRTS1 IN HRB1


+-----------------------------------------------------------------+
| There are pending changes related to the objects you are modifying.  |
| Use the "Display pending changes" option to see the pending changes. |
+-----------------------------------------------------------------+
```

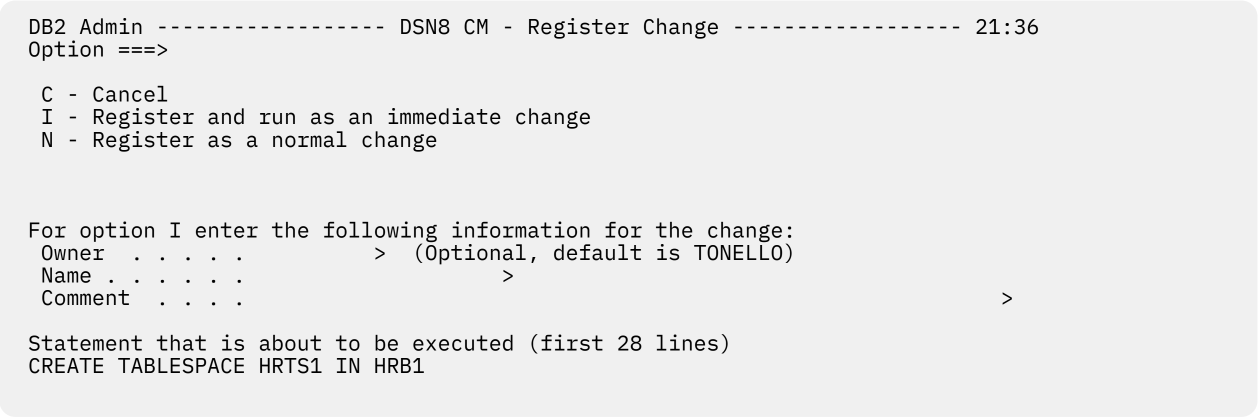
Figure 364. Example of Register Change panel (ADB2CMRG) when there are pending changes

Tip: Use option D to review the pending changes to help you make the appropriate register decision for your change.

If you register the change as an emergency change or as a normal change that supersedes the pending changes, any pending changes that are in ANALYZED status are set to DEFINED status. These changes will need to be analyzed again.

If you register the change as a normal change and ignore the pending changes, any pending changes that are in ANALYZED status are not set to DEFINED status.

The following figure shows an example of the **Register Change (ADB2CMRG)** panel if the affected objects have no pending changes:

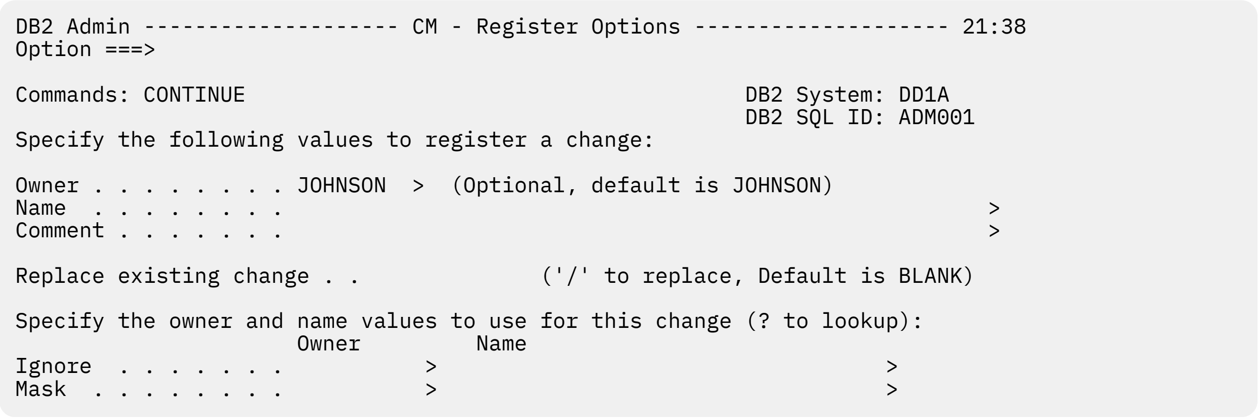


```
DB2 Admin ------------------ DSN8 CM - Register Change ------------------ 21:36
Option ===>

 C - Cancel
 I - Register and run as an immediate change
 N - Register as a normal change



For option I enter the following information for the change:
 Owner  . . . . .          >  (Optional, default is TONELLO)
 Name . . . . . .                    >
 Comment  . . . .                                                       >

Statement that is about to be executed (first 28 lines)
CREATE TABLESPACE HRTS1 IN HRB1
```

Figure 365. Example of Register Change panel (ADB2CMRG) when there are no pending changes

If you specify E or I on this panel to register the change as an emergency change (pending changes exist) or an immediate change (pending changes do not exist), you must specify an owner and a name for the change. An emergency or immediate change is run immediately.

6. Complete the fields on the **Register Options (ADB2CRO)** panel, and issue the CONTINUE command:



```
DB2 Admin -------------------- CM - Register Options -------------------- 21:38
Option ===>

Commands: CONTINUE                                      DB2 System: DD1A
                                                        DB2 SQL ID: ADM001
Specify the following values to register a change:

Owner . . . . . . . . JOHNSON  >  (Optional, default is JOHNSON)
Name  . . . . . . . .                                                    >
Comment . . . . . . .                                                    >

Replace existing change . .              ('/' to replace, Default is BLANK)

Specify the owner and name values to use for this change (? to lookup):
                      Owner         Name
Ignore  . . . . . . .          >                                  >
Mask  . . . . . . . .          >                                  >
```

Figure 366. Register Options panel (ADB2CRO)

Analyzing a change

After a change is registered, you must analyze it before you can run it. During this analyze step, Db2 Admin Tool analyzes how the change modifies existing objects and creates a work statement list (WSL) that can be used to run the change.

Before you begin

Before a change can be analyzed, it must be in DEFINED or ANALYZED status.

About this task

For more information about the analyze process, see “3. Analyze a change” on page 643.

Procedure

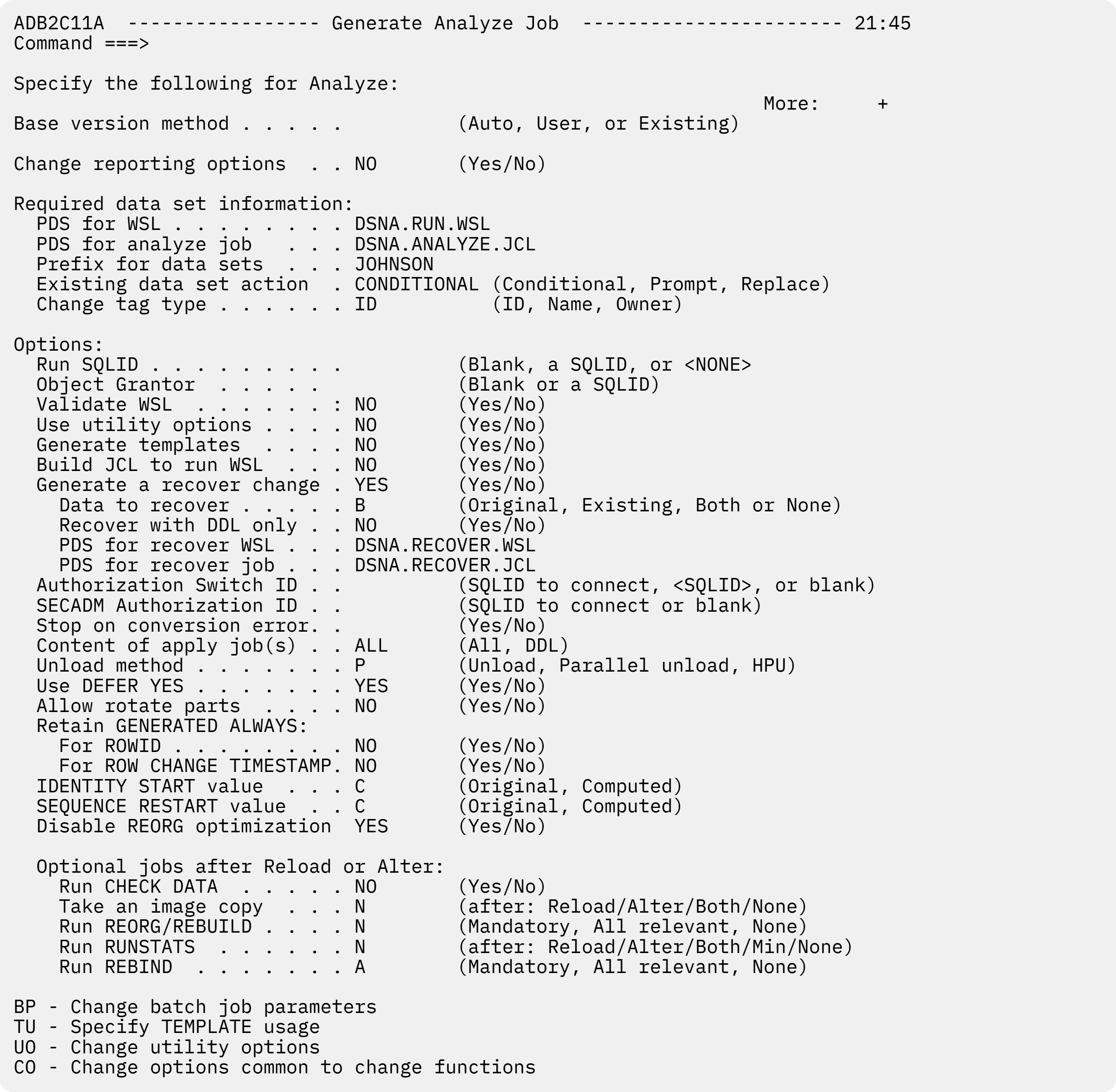
To analyze a change:

1. Navigate to the **Changes (ADB2C11)** panel by completing the following steps:
   a) On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
   b) On the **Change Management (CM) (ADB2C)** panel, specify option 1, and press Enter.

c) Optional: At the bottom of the **Manage Changes (ADB2C1)** panel, specify any search criteria to filter or limit the changes that are displayed.

d) On the **Manage Changes (ADB2C1)** panel, specify option 1, and press Enter.

2. On the **Changes (ADB2C11)** panel, issue the AN line command for the change that you want to analyze.
3. On the **Generate Analyze Job (ADB2C11A)** panel, complete the following fields:



```
ADB2C11A  ----------------- Generate Analyze Job  ----------------------- 21:45
Command ===>

Specify the following for Analyze:
                                                                  More:     +
Base version method . . . . .          (Auto, User, or Existing)

Change reporting options  . . NO       (Yes/No)

Required data set information:
  PDS for WSL . . . . . . . . DSNA.RUN.WSL
  PDS for analyze job   . . . DSNA.ANALYZE.JCL
  Prefix for data sets  . . . JOHNSON
  Existing data set action  . CONDITIONAL (Conditional, Prompt, Replace)
  Change tag type . . . . . . ID          (ID, Name, Owner)

Options:
  Run SQLID . . . . . . . . .          (Blank, a SQLID, or <NONE>)
  Object Grantor  . . . . .            (Blank or a SQLID)
  Validate WSL  . . . . . . : NO       (Yes/No)
  Use utility options . . . . NO       (Yes/No)
  Generate templates  . . . . NO       (Yes/No)
  Build JCL to run WSL  . . . NO       (Yes/No)
  Generate a recover change . YES      (Yes/No)
    Data to recover . . . . . B        (Original, Existing, Both or None)
    Recover with DDL only . . NO       (Yes/No)
    PDS for recover WSL . . . DSNA.RECOVER.WSL
    PDS for recover job . . . DSNA.RECOVER.JCL
  Authorization Switch ID . .          (SQLID to connect, <SQLID>, or blank)
  SECADM Authorization ID . .          (SQLID to connect or blank)
  Stop on conversion error. .          (Yes/No)
  Content of apply job(s) . . ALL      (All, DDL)
  Unload method . . . . . . . P        (Unload, Parallel unload, HPU)
  Use DEFER YES . . . . . . . YES      (Yes/No)
  Allow rotate parts  . . . . NO       (Yes/No)
  Retain GENERATED ALWAYS:
    For ROWID . . . . . . . . NO       (Yes/No)
    For ROW CHANGE TIMESTAMP. NO       (Yes/No)
  IDENTITY START value  . . . C        (Original, Computed)
  SEQUENCE RESTART value  . . C        (Original, Computed)
  Disable REORG optimization  YES      (Yes/No)

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

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

Base version method

Specify the base version method that Db2 Admin Tool is to use for the comparison to generate the changes. If you specify U (User-defined), you are prompted to specify the version scope to use. If you specify E (Existing), you are prompted to specify the base version to use. For more information about how to chose this value, see "Base version method for change analysis" on page 655.

Change reporting options

Specify whether to change reporting options before submitting the analyze job. If you specify YES, you are prompted to specify the reporting options to use.

Required data set information

Specify information about the data sets for the generated WSL and the generated jobs.

The value of the **Change tag type** option determines the PDS member names.

Tips:

- Keep the WSLs for Change Management changes separate from other WSLs. Do not mix them in the same data set.
- Choose a unique naming convention for the WSL and JCL data sets such that members for other Change Management databases are not put in the same data set.

Options

Specify job options. The following options are some of the preferences that you can specify:

Generate templates

Specify whether to use active templates. If you specify Yes, templates are generated for non-utility data sets with the template definitions that are defined for Object Comparison Tool. If you specify No, the value of the **Prefix for data sets** option is used to name the data sets. If the **Take an image copy** or **Run REORG** options are Yes, the utility templates are used.

Generate a recover change

Specify whether Db2 Admin Tool is to automatically generate a change that will recover the current change. If you specify Yes, you will be prompted to register the recover change.

Data to recover

If **Generate a recover change** = Yes, specify whether you want to recover the original data, the existing data, or both. *Original data* is the data that exists just before the original change is run. Original data can be recovered for objects that were unloaded during the original change. *Existing data* is the data that exists in the table just before the original change is recovered. If you specify that you want both, you can choose which data to recover at the time of recovery. You can also choose to recover only the DDL and not the data.

Optional jobs after Reload or Alter

Specify any optional utility job steps.

Recommendations:

- If the change affects tables that are part of an RI relationship, run the CHECK DATA utility.
- If the change is a pending ALTER operation, run the REORG utility.

Depending on the values that you specify, you might be prompted for additional information before the analysis jobs are generated and an ISPF Edit session is displayed.

4. If the change that you are analyzing was already analyzed (the change is in ANALYZED status), specify whether to continue or cancel the analyze request when you are prompted. If you continue, the change is placed back in DEFINED status before the new analyze job is created.
5. Edit and submit the generated analyze job.

   This job performs the analysis and generates a WSL for running the change.

   When the job completes successfully, the change is placed in ANALYZED status. If the job does not complete successfully, check the error messages in the job output. Correct any errors and then reanalyze the change by issuing the AN command.

   If you requested that a recover change be generated, the recover change is created and is also placed in ANALYZED status. A delta version for the recover change is also created.
6. Press PF3 to return to the **Changes (ADB2C11)** panel to verify that the status of the change is ANALYZED.

   If you requested that a recover change be generated, you can verify that it is also included in the list of changes.

   Tip: If you return to the **Changes (ADB2C11)** panel before the submitted job completes, you can click the REFRESH command after the job completes to see the refreshed status of the change.

What to do next

"Running a change" on page 656

Related concepts

"Change Management terminology" on page 639
Before you use Change Management (CM), you should understand the terms that CM uses.

Related reference

"Types of changes and change status" on page 644
To facilitate change management, Db2 Admin Tool categorizes changes into several types and assigns a status to each change as it moves through the change management process.

Base version method for change analysis

During the analysis of a change, Db2 Admin Tool needs to know the current state of the objects that are being changed.

Db2 Admin Tool can get this information from an existing version that was created earlier or extract the information from the Db2 catalog. When the information is extracted from the Db2 catalog, Db2 Admin Tool either extracts it based on a user-defined scope or based on the objects that are being changed.

Specifically, Db2 Admin Tool generates the base version using one of the following methods:

Automatic (A)
The base version is generated from the Db2 catalog using the objects that are referenced in the change.

User-defined (U)
The base version is generated from the Db2 catalog using the objects that are specified in the version scope.

Existing (E)
An existing base version is used. Db2 Admin Tool uses the current contents of the existing version and the contents of the Db2 catalog are not considered.

If no prerequisite changes exist for the change, you can specify which of the preceding methods Db2 Admin Tool uses. If prerequisite changes exist, Db2 Admin Tool chooses the method based on the following criteria:

User-defined (U)
This method is forced if all of the prerequisites have a status of ANALYZED and use the same version scope. The same version scope will be used for the change you are analyzing.

Existing (E)
This method is forced if all of the prerequisites have a status of ANALYZED and use the same base version. The same base version will be used for the change you are analyzing.

Automatic (A)
This method is forced if neither of the previous conditions are true.

The base version method that you choose depends on your installation's needs. Your shop might prefer to create a new snapshot (base version) after every change to use as a backup and also as the base version for new changes. When the next change needs to be analyzed, you can specify to have the existing version used and avoid extracting the object definitions from the Db2 catalog to get the current status. Processing time is saved when you do not have to extract the objects from the catalog.

Other shops might want to work on one application at a time. A scope can be defined that includes all of the objects in the application (for example, one or more databases) and always use this scope as the base when analyzing a change.

Some shops might not want to use existing base versions or user-defined scopes and choose to have the base automatically generated from the Db2 catalog when analyzing a change.

Running a change

When you run a change, the work statement list (WSL) that was created during the analyze process is run.

Before you begin

A change must be in ANALYZED status before you can run it.

Also, if you plan to create a base version of the objects after the change, ensure that you have a version scope that defines the set of objects to be included in the base version.

Tip: If you are running a change for an object that is outside of the current version scope, update the definition of the version scope. The version scope should include all objects so that any subsequent changes for which you create a new base version includes these objects. For example, if the version scope includes database DB01 and the change is to add a second database DB02, change the definition of the version scope to include database DB02.

Optional: Changes might have been made to the Db2 catalog since the WSL was generated during the analyze process. (These changes can include changes that were made outside of Change Management.) As a result, that WSL might now conflict with or undo those changes. To minimize the possibility of run-time errors, you can verify the WSL by generating a new WSL and comparing it to the WSL that was generated during the normal analyze process.

About this task

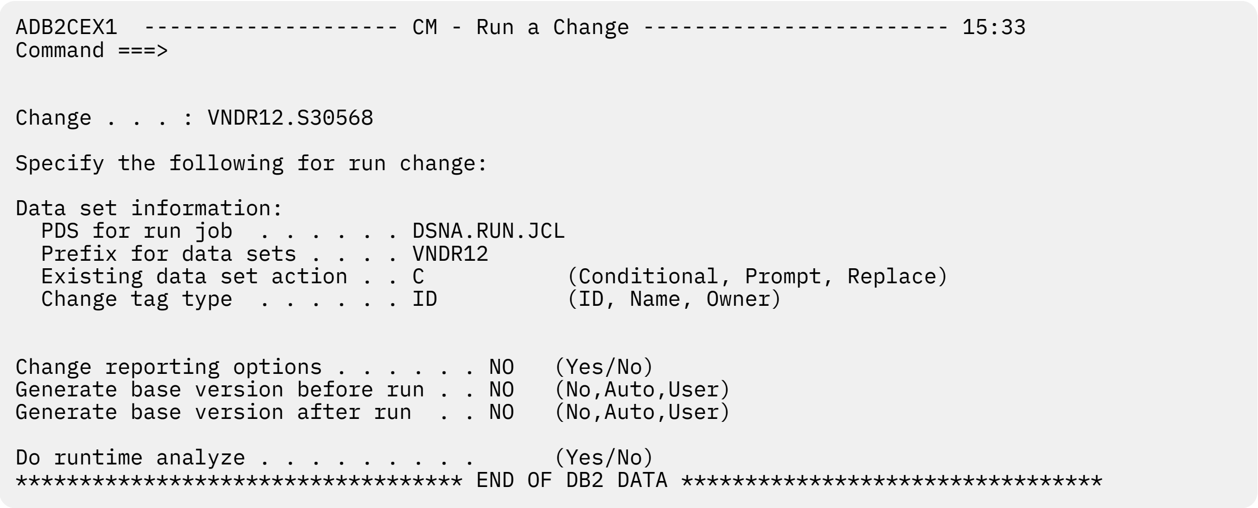
For more information about the run process, see "4. Run a change" on page 644.

Procedure

To run a change:

1. Navigate to the **Changes (ADB2C11)** panel by completing the following steps:
   a) On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
   b) On the **Change Management (CM) (ADB2C)** panel, specify option 1, and press Enter.
   c) Optional: At the bottom of the **Manage Changes (ADB2C1)** panel, specify any search criteria to filter or limit the changes that are displayed.
   d) On the **Manage Changes (ADB2C1)** panel, specify option 1, and press Enter
2. On the **CM - Changes (ADB2C11)** panel, enter the RN line command next to the change that you want to run

   If the change has prerequisite changes, you are prompted to run the prerequisite changes first.
3. On the **Run a Change (ADB2CEX1)** panel, complete the following fields, and press Enter:



```
ADB2CEX1  -------------------- CM - Run a Change ------------------------ 15:33
Command ===>


Change . . . : VNDR12.S30568

Specify the following for run change:

Data set information:
  PDS for run job  . . . . . . DSNA.RUN.JCL
  Prefix for data sets . . . . VNDR12
  Existing data set action . . C           (Conditional, Prompt, Replace)
  Change tag type  . . . . . . ID          (ID, Name, Owner)


Change reporting options . . . . . . NO   (Yes/No)
Generate base version before run . . NO   (No,Auto,User)
Generate base version after run  . . NO   (No,Auto,User)

Do runtime analyze . . . . . . . . .      (Yes/No)
*********************************** END OF DB2 DATA ********************************
```

Data set information:
Specify information about the data set that you want to use for the generated job.

Change reporting options:
Specify whether to change the Object Compare reporting options for the runtime analyze.

Generate base version before run:
Specify whether to generate a new base version immediately before the change is implemented. You can specify one of the following options:

No
A new base version is not created.

Auto
A new base version is created. The objects in the base version are automatically determined by the product based on the objects that are being changed.

User
A new base version is created. The objects in the base version are the ones that are listed in the user-specified version scope.

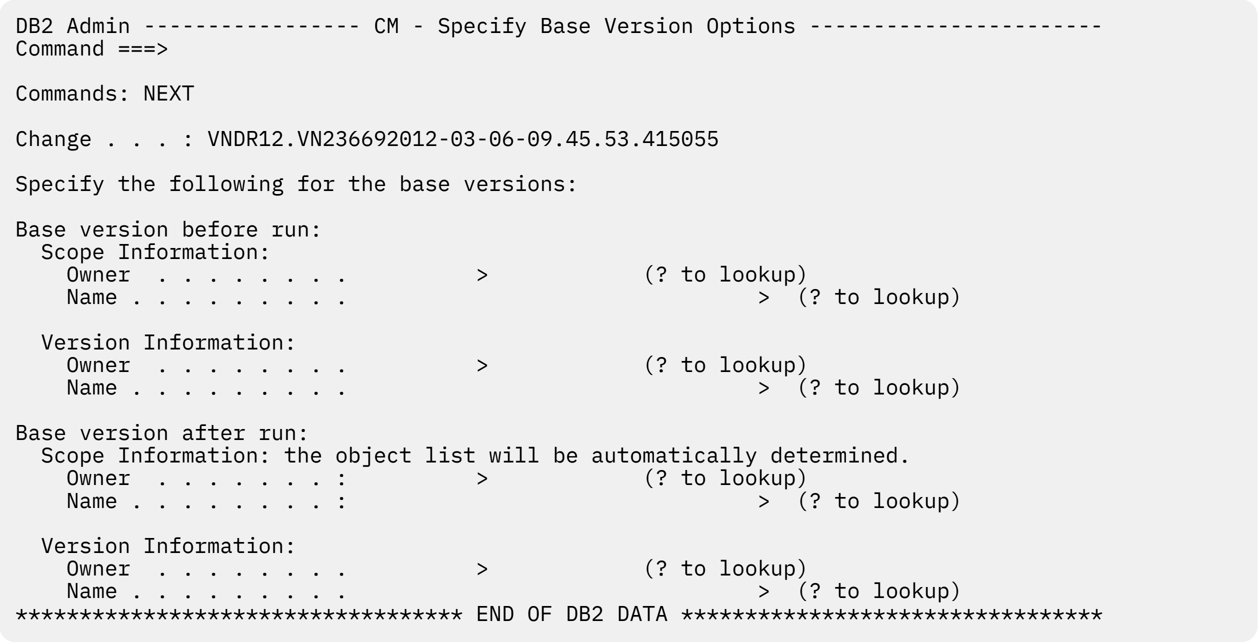
Generate base version after run:
Specify whether to generate a new base version immediately after the change is implemented. You can specify one of the following options:

No
A new base version is not created.

Auto
A new base version is created. The objects in the base version are automatically determined by the product based on the objects that are being changed.

User
A new base version is created. The objects in the base version are the ones that are listed in the user specified version scope.

If you specify `Auto` or `User` for **Generate base version before run** or **Generate base version after run**, the **Specify Base Version Options (ADB2CEX3)** panel is displayed to prompt you for the name of the new base version. If you specify `User`, you are also promoted for the name of the version scope.



```
DB2 Admin ----------------- CM - Specify Base Version Options ----------------------
Command ===>

Commands: NEXT

Change . . . : VNDR12.VN236692012-03-06-09.45.53.415055

Specify the following for the base versions:

Base version before run:
  Scope Information:
    Owner  . . . . . . . .          >            (? to lookup)
    Name . . . . . . . . .                                >  (? to lookup)

  Version Information:
    Owner  . . . . . . . .          >            (? to lookup)
    Name . . . . . . . . .                                >  (? to lookup)

Base version after run:
  Scope Information: the object list will be automatically determined.
    Owner  . . . . . . . :          >            (? to lookup)
    Name . . . . . . . . :                                >  (? to lookup)

  Version Information:
    Owner  . . . . . . . .          >            (? to lookup)
    Name . . . . . . . . .                                >  (? to lookup)
*********************************** END OF DB2 DATA *********************************
```

4. Edit and submit the generated job.
5. Press PF3 to return to the **CM - Changes (ADB2C11)** panel and verify that the status of the change is COMPLETE.

Tip: If you return to the **Changes (ADB2C11)** panel before the submitted job completes, you can click the REFRESH command after the job completes to see the refreshed status of the change.

What to do next

If the job fails (the status of the job does not change to COMPLETE), the action to take depends upon the current status of the change:

- If the status is ANALYZED, check the job output. If a message indicates that the run-time WSL did not match the WSL that was generated during the analyze process, return to the **CM - Changes (ADB2C11)** panel and issue the AN line command to reanalyze the change. Then, run the change again.

  When you run a change, a new run-time WSL is generated for the change and its recover change, if one exists, based on the current Db2 catalog and using the automatic base version method. The run-time WSL files are compared with the WSL files that were created during the normal analyze process. If the DDL and DCL content are the same, the run job continues and the WSL that was generated during the normal analyze is run to apply the changes. If the DDL or DCL content are different, the run job stops with a return code of 8, and the change is not run. In this case, you need to reanalyze the change to resolve the differences in the WSLs.
- If the status is RUNNING, check the job output. Determine the cause of the failure and make any necessary corrections. Then, return to the **CM - Changes (ADB2C11)** panel, issue the ER line command to edit the run job, and resubmit it. When you submit the run job, the job is restarted at the appropriate step.

  When you issue the ER line command, the JCL for the run job is placed in edit mode. Before the job is displayed in edit mode, a RESTART parameter is automatically added to the job card to restart the job at the step that runs ADBTEP2 so that you do not have to determine the step name where the job should be restarted. In addition, if the RESTART parameter for ADBTEP2 was changed to RESTART(NO) by using the ER line command during an earlier edit session, the parameter is automatically changed to RESTART(YES), because ADBTEP2 must be restarted with the parameter RESTART(YES). If the parameter is missing, ADBTEP2 assumes a YES value.

  These automatic changes and any edit changes that you make are saved to the JCL data set so that you do not need to re-enter the changes for a subsequent ER line command for the job.

  Important: Any user can use the ER line command to edit and resubmit a change in RUNNING status. The user who originally ran the change is not required to resubmit the job. The restart record in the checkpoint table for the change retains the userid of the original submitter. Db2 Admin Tool locates the record by using the CHANGEID parameter. The RN and ER line commands automatically include the CHANGEID parameter when the run job is built so that you do not have to manage this process.
- If the status is DEFINED and message ADB9356E was returned, you need to reanalyze the change. ADB9356E identifies the change that supersedes this change and is responsible for setting the status to DEFINED. Return to the **CM - Changes (ADB2C11)** panel and issue the AN line command to reanalyze the change.

Related concepts

“Change Management terminology” on page 639
Before you use Change Management (CM), you should understand the terms that CM uses.

Related tasks

“Analyzing a change” on page 652
After a change is registered, you must analyze it before you can run it. During this analyze step, Db2 Admin Tool analyzes how the change modifies existing objects and creates a work statement list (WSL) that can be used to run the change.

Related information

“ADB9356E” on page 1176

The change was superseded by change *ID*(*owner*.*name*) on *timestamp* and needs to be re-analyzed.

Recovering a table if the change fails

If a table change fails and the original table is dropped, you can restore the table to its original state.

Procedure

To recover a table if the change fails:

1. Drop the new table if it was created.
2. Re-create the original table using the extracted DDL.
3. Load the table by using the unload data set. If other tables exist in the table space, specify the RESUME YES option in the LOAD utility statement .
4. Create a new image copy of the table space.
5. Run the RUNSTATS utility on the table.

Managing Changes by using the CM batch interface

The Change Management (CM) batch interface is a JCL procedure that enables you to create, customize, and reuse batch jobs when managing changes in Db2 Admin Tool Change Management. You can import, analyze, run, and recover changes by submitting batch jobs without using the Change Management ISPF panels.

Before you begin

Before invoking the CM batch interface, take the following actions:

- Before you can run any Db2 for z/OS online utilities in CM batch, you must have one of the following Db2 stored procedures set up and available:
  - DSNUTILV (for Db2 12 for z/OS)
  - DSNUTILU (for prior versions of Db2 for z/OS)

  By default, CM batch uses these stored procedures to run utilities. You can change that behavior by using the CM batch parameter use_dsnutil_sp.
- Optionally configure the CM batch interface by taking one or more of the following actions:
  - "Defining your own JCL symbols as parameters to the CM batch interface" on page 660
  - "Customizing the Change Management (CM) batch interface JCL procedure name" on page 661

About this task

The CM batch interface can be used to manage a change that was created with the Db2 Admin Tool panels. Likewise, a change that was imported using the CM batch interface can be managed using Db2 Admin Tool panels.

Specifically, the following Change Management functions are supported by the CM batch interface:

- Run compare (invokes IBM Db2 Object Comparison Tool for z/OS to generate a delta change file that can be managed by Db2 Admin Tool Change Management)
- Import a mask
- Import an ignore
- Import a change (equivalent to using register change in the Db2 Admin Tool panels), including importing one or more DDL or delta change files into a single change
- Analyze a change (using the automatic base version method)
- Analyze a multi-target change
- Build a run job

- Run a change
- Recover a change

Restriction: The following Change Management functions are not supported in the CM batch interface:

- Report changes
- Import a version file
- Import a version scope
- Analyze a change (using the user-defined or existing base version method)

Procedure

To perform CM functions by using the CM batch interface:

- Invoke the JCL procedure to enable the CM batch interface.

  The default CM batch interface JCL procedure name is GOCCM.

  Example:

  ```
  //CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
  ```

  You can use the same JCL procedure for single or multiple Db2 subsystems.
- Call any of the supported CM functions. More than one of these functions can be performed in a single call to the CM batch interface, except for the "recover change" function, which cannot be performed with any other action.
- Specify CM batch interface parameters as needed.

  For more information about parameters, including syntax, see “Parameters for CM batch interface” on page 662. For a complete list of parameters, see “CM batch parameter definitions” on page 664.
- If you use the CM batch interface to import a DDL file, the first line of the DDL file must be a simple SQL comment. A simple SQL comment starts with two dash symbols (`--`). Otherwise, import change errors might occur.

Example

For examples of invoking the CM batch interface, see “Examples: Invoking the Change Management (CM) batch interface for various actions” on page 810.

Related concepts

“Base version method for change analysis” on page 655
During the analysis of a change, Db2 Admin Tool needs to know the current state of the objects that are being changed.

Related information

Creating a Change Management batch job to run compare (IBM Db2 Object Comparison Tool for z/OS 12.1.0)

Defining your own JCL symbols as parameters to the CM batch interface

Because Change Management (CM) batch interface is a JCL procedure, it provides the flexibility for you to define parameters with your own JCL symbols.

About this task

When you invoke CM batch interface, you use a JCL EXEC statement, such as:

```
//GOCCM  EXEC GOCCM,SSID=DSNA,PLAN=ADB
```

You can make customizations such that your own JCL symbols are allowed on this EXEC statement.

Procedure

To define your own JCL symbols as parameters to the CM batch interface:

1. Add the symbols to the CM batch interface JCL procedure:

   Example: In the following snippet of member GOCCM, the user-defined JCL symbols on the PROC statement are P1 and P2, and the values of P1 and P2 are specified in PARM.

   ```
   //GOCCM  PROC SSID=,PLAN=,P1=,P2=
   //GOCCM  EXEC PGM=IKJEFT01,DYNAMNBR=200,
   //  PARM=('CALL *(GOCCCM) ''/SSID(&SSID) PLAN(&PLAN) '
   // 'P1=&P1 P2=&P2',
   //  '''')
   <snip>
   //GOCCM  PEND
   ```

2. When you invoke the CM batch interface JCL procedure, include the new parameters on the EXEC statement.

   For example, the following EXEC JCL statement invokes the CM batch interface to analyze and build a run job for a change:

   ```
   //GOCCM  EXEC GOCCM,SSID=DSNA,PLAN=ADB,P1=ABC,P2=XYZ
   ```

Results

Because you added your own JCL symbols to PARM, any JCL job that CM batch interface creates contains the added JCL symbols on the EXEC statement. For example, the EXEC JCL statement generated by CM batch interface in the run job contains the values for P1 and P2, as follows:

```
//GOCCM  EXEC GOCCM,SSID=DSNA,PLAN=ADB,P1=ABC,P2=XYZ
```

Customizing the Change Management (CM) batch interface JCL procedure name

The default CM batch interface JCL procedure name is GOCCM. If you use a member name other than GOCCM, you must customize the CM batch interface JCL procedure.

About this task

A cataloged procedure name is a member or alias of a PDS or PDSE that is defined in your environment to be the libraries that store cataloged JCL procedures.

Procedure

To customize the CM batch interface JCL procedure name:

1. Set the MBR parameter in PARM to the member name that you want to use.

   Example: Suppose the CM batch interface JCL procedure is stored in member TEST01 instead of the default GOCCM. In this case, in member TEST01, set the MBR parameter in PARM to TEST01, as shown in the following example:

   ```
   //GOCCM  PROC SSID=,PLAN=,P1=,P2=
   //GOCCM  EXEC PGM=IKJEFT01,DYNAMNBR=200,
   //  PARM=('CALL *(GOCCCM) ''/SSID(&SSID) PLAN(&PLAN) ',
   // 'MBR=TEST01 P1=&P1 P2=&P2',
   //  '''')
   <snip>
   //GOCCM  PEND
   ```

2. To invoke the CM batch interface using your cataloged JCL procedure, the EXEC statement must include your procedure name.

Example: The following EXEC JCL statement invokes the CM batch interface using the TEST01 cataloged JCL procedure to analyze and build a run job for a change:

```
//GOCCM  EXEC TEST01,SSID=DSNA,PLAN=ADB,P1=ABC,P2=XYZ
```

Results

The EXEC JCL statement generated by the CM batch interface in the run job uses the TEST01 cataloged procedure, as follows:

```
//GOCCM  EXEC TEST01,SSID=DSNA,PLAN=ADB,P1=ABC,P2=XYZ
```

Parameters for CM batch interface

The Change Management (CM) batch interface includes a list of parameters that enable you to control various aspects of managing changes, including what action the CM batch interface performs when called.

These parameters enable you to customize various aspects of managing a change, such as:

- Prefixes for data sets that are dynamically created by the CM batch interface
- The name of the PDS to store work statement list (WSL) files
- The name of the PDS to store JCL run jobs for running changes
- The default "change owner" name to use when creating a new change
- The default "change name" to use when creating a new change
- Analyze reporting options
- Admin templates
- Utility options

  Requirement: To specify values for utility options, you must set the “USE_UTILITY_OPTIONS” on page 729 parameter to Y. Otherwise, default values are used.

For a full list of CM batch interface parameters, see “CM batch parameter definitions” on page 664.

Action parameters

You can use the following parameters to control what action the CM batch interface performs.

Table 35. Action parameters for CM batch interface

Action	**Parameter name**
Run compare	“ACTION_COMPARE” on page 666
Analyze a change	“ACTION_ANALYZE_CHANGE” on page 665
Build a run job	“ACTION_BUILD_RUN_JOB” on page 666
Generate DDL	“ACTION_GENERATE_DDL_FROM_BASE_VERSION” on page 669
Generate a base version	“ACTION_GENERATE_BASE_VERSION” on page 669
Generate JCL from a WSL	“ACTION_GENERATE_JCL_FROM_WSL” on page 670
Generate a WSL	“ACTION_GENERATE_WSL” on page 670
Run a WSL	“ACTION_RUN_WSL” on page 672
Import a change	“ACTION_IMPORT_CHANGE” on page 670
Export changes into a delta change file	“ACTION_EXPORT_CHANGE” on page 668
Import an ignore	“ACTION_IMPORT_IGNORE” on page 671
Import a mask	“ACTION_IMPORT_MASK” on page 671
Delete a mask	“ACTION_DELETE_MASK” on page 668
Run a change	“ACTION_RUN_CHANGE” on page 672
Cancel a change	“ACTION_CANCEL_CHANGE” on page 666

Table 35. Action parameters for CM batch interface (continued)

Action	Parameter name
Delete a change	"ACTION_DELETE_CHANGE" on page 667
Recover a change	"ACTION_RECOVER_CHANGE" on page 671
Convert a WSL to ISPF table format	"ACTION_CONVERT_TO_ISPF_WSL" on page 667
Convert a WSL to a readable data set	"ACTION_CONVERT_TO_READ_WSL" on page 667

Parameter syntax

Use the following syntax rules when specifying CM batch parameters:

- **Use of quotation marks:** Enclose the value of the parameter in single quotation marks, as follows:

  ```
  parameter_name = 'parameter_value'
  ```

  When specifying the fully qualified PDS name, enclose the PDS name in double quotation marks within single quotation marks. For example, when the WSL PDS is named HLQ.BATCH.WSL specify the following:

  ```
  PDS_FOR_WSL='''HLQ.BATCH.WSL'''
  ```

- **Use of uppercase and lowercase:** Most parameter values are not case sensitive. However, the following parameter values are case sensitive:
  - symbol parameters
  - parameters related to data set names
  - parameters related to an object owner, name, or comment
- **Defining a user symbol:** When defining a user specified symbol by using the symbol_name and symbol_value parameters, the symbol name must begin with an ampersand (&) and end with a period (.). A semi-colon must be specified immediately after the symbol value, as follows:

  ```
  symbol_name = '&TASK#.' symbol_value = 'ABC';
  ```

- **Specifying a fully qualified data set name:** When specifying a fully qualified data set name, you can either use two single quotation marks to represent one single quotation mark, or wrap the parameter value in double quotation marks.

  For example, the following specification results in a WSL data set name of `WALDO1.WALDO2.WSL`:

  ```
  prefix_for_data_sets = 'WALDO1'
  pds_for_wsl = 'WALDO2.WSL'
  ```

  If you want the WSL data set name to be `WALDO2.WSL`, specify one of the following:

  -
    ```
    prefix_for_data_sets = 'WALDO2'
    pds_for_wsl = 'WSL'
    ```

  -
    ```
    prefix_for_data_sets = 'WALDO1'
    pds_for_wsl = '''WALDO2.WSL'''
    ```

Db2 Admin Tool data set templates

You can use Db2 Admin Tool data set templates to override the default values for some data sets that are used to process a change.

To do so, use the admin_dataset_type parameter to specify the type of data set whose attributes you want to overwrite. For a list of possible data set types that you can specify, see "ADMIN_DATASET_TYPE" on page 684.

After the admin_dataset_type parameter, specify one or more of the following data set template parameters to override certain data set attributes:

- admin_dataset_bufno
- admin_dataset_dataclas
- admin_dataset_device_unit
- admin_dataset_dir
- admin_dataset_dsn
- admin_dataset_dsntype
- admin_dataset_expdt
- aadmin_dataset_maxvol
- admin_dataset_mgmtclas
- admin_dataset_retpd
- admin_dataset_space_priqty
- admin_dataset_space_secqty
- admin_dataset_space_type
- admin_dataset_storclas
- admin_dataset_type
- admin_dataset_volume

End the group of parameters (admin_dataset_type and the other data set template parameters) with a semicolon (;), as shown in the following example:

```
admin_dataset_type = 'CHG'
admin_dataset_dsn = 'CHG.T&TIME.'
admin_dataset_space_priqty = '20';
```

You can specify multiple groups of these parameters. Each group is separated by a semicolon.

Related tasks

“Setting default CM batch parameter values by using profiles” on page 790
You can establish and maintain your own default values for CM batch parameters. Individual invocations of CM batch can override these default values as needed.

Related information

Creating a Change Management batch job to run compare (IBM Db2 Object Comparison Tool for z/OS 12.1.0)

CM batch parameter definitions

You can use Change Management (CM) batch interface parameters to control Change Management (CM) actions and settings.

For the parameters that control utility options, see “Utility option parameters” on page 745. All other CM batch parameters are included in the following list in alphabetical order.

Note: If a parameter has two values listed together, such as **YES or Y**, the second value (in this case Y) is the short value. Either value is acceptable; they both have the same meaning.

ACCELERATOR_LOAD_LOCKMODE

Specifies the lock mode for the SYSPROC.ACCEL_LOAD_TABLES procedure for accelerated tables.

Values:

TABLESET
TABLE
ROW

PARTITIONS
NONE

Default:
TABLESET

For details on the possible values of lock mode, see the information about *lock_mode* in SYSPROC.ACCEL_LOAD_TABLES (IBM DB2 Analytics Accelerator for z/OS 7.5).

ACCEPT_FL

Specifies the maximum Db2 version function level (FL) with which the CM batch job is allowed to continue.

If the specified value is greater than the function level that is supported by Db2 Admin Tool, a warning message is issued.

If the specified value is less than the current function level value of the Db2 subsystem, an error message is issued and the CM batch job is terminated.

Values:
An integer value in the range 501 - 999.

Default:
blank

A blank value means that CM batch jobs are not allowed to run if the current function level of the Db2 subsystem is greater than the function level that is supported by Db2 Admin Tool.

Related information:

Adopting new capabilities in Db2 12 continuous delivery (Db2 12 for z/OS)
"Support for Db2 continuous delivery" on page 69

ACTION_ANALYZE_CHANGE

Specifies whether to analyze a change. If a change is also being imported, the change that is analyzed is the newly imported change. Otherwise, the change to be analyzed is identified by the **CHANGE_OWNER** and **CHANGE_NAME** parameters.

Values:

Y
Specifies that the change identified by the **CHANGE_OWNER** and **CHANGE_NAME** parameters is analyzed. If a change is also being imported, the change that is analyzed is the change identified by the **NEW_CHANGE_OWNER** and **NEW_CHANGE_NAME** parameters. A change that is already in ANALYZED state is reanalyzed.

N
Specifies that no change is analyzed.

C
Same as the **Y** parameter value except that the analyze is not done if the change is already in the ANALYZED state. For the **C** parameter, return codes of 0 and 1 are defined as follows:

- 0 indicates that analyze is not done because the change is already in the ANALYZED state.
- 1 indicates that analyze is done, and no warnings or errors are issued. The state is the same the state of `ACTION_ANALYZE_CHANGE = 'Y'` with RC=0.

There are no changes to the meanings of other RC values.

blank
Specifies that this parameter defaults to Y if a change is imported during this call to the Change Management batch interface.

Default:
blank

ACTION_BUILD_RUN_JOB

Specifies whether a run job is created for a change. If a change is also being imported, a run job is created for the newly imported change. Otherwise, a run job is created for the change identified by the **CHANGE_OWNER** and **CHANGE_NAME** parameters.

Values:

Y

Specifies that a run job is created for the change identified by the **CHANGE_OWNER** and **CHANGE_NAME** parameters. If a change is also being analyzed, a run job is created for the change after it is analyzed.

N

Specifies to not create a run job for the change.

blank

Specifies that this parameter should default to Y if a change is analyzed during this call to the Change Management (CM) batch interface.

Default:

blank

ACTION_CANCEL_CHANGE

Specifies whether to cancel a change that is specified by the **CHANGE_OWNER** and **CHANGE_NAME** parameters. No other CM Batch actions are allowed when you are requesting a change to be canceled. Any type of change can be canceled except for a multi-target change and a change that is in COMPLETE state.

Values

U

Specifies an unconditional cancel change. The specified change is canceled even if other changes have the change as a prerequisite. If other changes have the change as a prerequisite, the changes that depend on the change that is being canceled are set to DEFINED status and must be analyzed before being run. A list of changes that have the change to be canceled as a prerequisite are listed.

C

Specifies a conditional cancel change. The specified change is canceled if no other changes have the change as a prerequisite change. If other changes have the change as a prerequisite change, an error message is issued. A list of changes that have the change to be canceled as a prerequisite are listed.

N

Specifies to not cancel the change.

Default

N

ACTION_COMPARE

Specifies whether to run the IBM Db2 Object Comparison Tool for z/OS to define a change that can be imported and managed by Db2 Admin Tool Change Management.

Values

Y

Specifies to run Db2 Object Comparison Tool to define a change that can be managed by Db2 Admin Tool Change Management. A compare report and a delta change file is generated that can be imported as a new change. The delta change file attributes are taken from the parameters for **ADMIN_DATASET_TYPE** = 'DELTA'.

By specifying **ACTION_COMPARE** = 'Y' and **ACTION_IMPORT_CHANGE** = 'N', you can run Db2 Object Comparison Tool to just generate a compare report and delta change file, without importing the result as a change. This setting enables you to view the differences between the compare source and target, and perhaps run the compare multiples times to fine-tune the differences between the source and target. When no more compares are needed and the change is ready to be deployed, the delta change file can be imported as a new change.

The files with DD names that start with IMCHG are not used.

N

Specifies to not run Db2 Object Comparison Tool to define the change.

Default

N

Related information:

Creating a Change Management batch job to run compare (IBM Db2 Object Comparison Tool for z/OS 12.1.0)

ACTION_CONVERT_TO_ISPF_WSL

Specifies whether the text WSL given by the CM batch parameters will be converted to an ISPF table format. This parameter is supported by CAF.

Values:

Y

Converts the readable work statement list (WSL) to an ISPF table. The readable WSL is specified by the PDS_FOR_WSL_CONV and WORKLIST_NAME_CONV parameters. The ISPF WSL is specified by the PDS_FOR_WSL and WORKLIST_NAME parameters.

N

Does not convert the WSL.

Default:

N

ACTION_CONVERT_TO_READ_WSL

Specifies whether the WSL that is given by the CM batch parameters will be converted to a readable data set. This parameter is supported by CAF.

Values:

Y

The WSL will be converted to a readable data set that is specified by the parameters PDS_FOR_WSL_CONV and WORKLIST_NAME_CONV.

N

Does not convert the WSL.

Default:

blank

ACTION_DELETE_CHANGE

Specifies whether to delete a change specified by the **CHANGE_OWNER** and **CHANGE_NAME** parameters. No other CM Batch actions are allowed when you are requesting a change to be deleted. Any type of change can be deleted except for a multi-target change.

Values

C

Specifies a conditional delete change. The specified change is deleted if no other changes have the change as a prerequisite. If other changes have the change as a prerequisite, an error message is issued. A list of changes that have the change to be deleted as a prerequisite are listed.

N

Specifies to not delete the change.

U

Specifies an unconditional delete change. The specified change is deleted even if other changes have the change as a prerequisite. If other changes have the change as a prerequisite, the changes that depend on the change that is being deleted are set to DEFINED status and must be analyzed before being run. A list of changes that have the change to be deleted as a prerequisite are listed.

Default

N

ACTION_DELETE_MASK

Specifies whether to delete the mask specified by the **COMPARE_MASK_OWNER** and **MASK_NAME** parameters. No other CM Batch actions are allowed when you are requesting a mask to be deleted.

Values

C

Delete the mask if it is not associated with a registered change that needs the mask for implementation. If the mask is needed by one or more changes for implementation, the names of changes are displayed and the mask is not deleted.

CONDITIONAL mode does not cover the scenario in which changes can have a mask associated with it but the mask is no longer needed to implement the change. For example, when a change is imported using masking, after the import completes the mask is no longer needed to implement that change.

N

The delete mask action is not enabled.

U

Delete the mask even if it is associated with a registered change that needs the mask for implementation. The names of changes that need the mask for implementation are displayed and the mask is deleted.

UNCONDITIONAL mode does not cover the scenario in which changes can have a mask associated with it but the mask is no longer needed to implement the change. For example, when a change is imported using masking, after the import completes the mask is no longer needed to implement that change. In this scenario, UNCONDITIONAL does not report the imported change.

Default

N

ACTION_EXPORT_CHANGE

Specifies whether to export the indicated changes into a delta change file. The changes to export are identified by using the **CHANGE_OWNER** and **CHANGE_NAME** parameters. Each pair of **CHANGE_OWNER** and **CHANGE_NAME** parameters must end with a semi-colon. For example, to export changes A.C1 and A.C2, in this order, the following set of parameters can be specified:

```
//PARMS DD *

ACTION_EXPORT_CHANGE = 'ASIS'

CHANGE_OWNER = 'A'

CHANGE_NAME = 'C1';

CHANGE_OWNER = 'A'

CHANGE_NAME = 'C2';
```

The delta change file is created according the template for `ADMIN_DATASET_TYPE = 'DELTA'`. For more information about these templates, see "Db2 Admin Tool data set templates" on page 663 .

Values

ASIS

The specified changes are exported in the order that is specified by the **CHANGE_OWNER** and **CHANGE_NAME** parameters.

PREREQ

Db2 Admin Tool analyzes the specified changes for prerequisites and includes any prerequisite changes for the listed changes.

NO

The changes are not exported.

Default

NO

ACTION_GENERATE_BASE_VERSION

Specifies whether and how to generate a base version. This parameter enables you to start Change Management (CM) batch interface only to generate a base version. The **GENERATE_BASE_VERSION_BEFORE_RUN** and **GENERATE_BASE_VERSION_AFTER_RUN** parameters enable you to configure Change Management (CM) batch interface so that base versions are automatically generated during the run change process. Generating a base version by using the **ACTION_GENERATE_BASE_VERSION** parameter and choosing to generate DDL from a base version (**GENERATE_DDL_FROM_BASE_VERSION** parameter) in the same invocation of Change Management (CM) batch interface enables to you save the current definitions of objects in the base version and also to generate a DDL file from these object definitions.

Values:

AUTO

A base version is generated and the content is automatically determined by the product for the specified change entry. The content of the base version is based on the registered change statements for the specified change entry.

USER

A base version is generated and the content is determined by a user-specified version scope.

NO

A base version is not generated. However, this setting has no control over whether a base version is automatically generated as determined by the **GENERATE_BASE_VERSION_BEFORE_RUN** and **GENERATE_BASE_VERSION_AFTER_RUN** parameters.

Default:

NO

ACTION_GENERATE_DDL_FROM_BASE_VERSION

Specifies whether to generate DDL from a base version. The generated file must be run by using the Db2 Admin Tool ADBTEP2 program. See ADBTEPR SAMP member for a sample job of running ADBTEP2.

Values:

BEFORE_RUN

DDL and Db2 Admin Tool statements are generated for the base version that was created before the specified change was implemented.

AFTER_RUN

DDL and Db2 Admin Tool statements are generated for the base version that was created after the specified the change was implemented.

SOURCE

DDL and Db2 Admin Tool statements are generated for the base version that is recorded as the source base version for the specified change.

TARGET

DDL and Db2 Admin Tool statements are generated for the base version that is recorded as the target base version for the specified change.

USER

DDL and Db2 Admin Tool statements are generated for the user-specified base version that is identified by the **BASE_VERSION_OWNER** and **BASE_VERSION_NAME** parameters.

NO

DDL and Db2 Admin Tool statements are not generated for any base version.

Default

NO

ACTION_GENERATE_JCL_FROM_WSL

Generate JCL based on an existing WSL or a WSL that is being generated in the same batch job by Db2 Object Comparison Tool.

Values:

Y

Generate JCL for one of the following WSLs:

- An existing WSL. In this case, you must specify the PDS_FOR_WSL parameter and the following parameter values:

```
ACTION_COMPARE = 'N'
ACTION_GENERATE_WSL = 'N'
```

- A WSL that is being generated while running a comparison that does not use Change Management. This action is indicated by the following parameter values:

```
ACTION_COMPARE = 'Y'
ACTION_GENERATE_WSL = 'Y'
```

In this case, the PDS_FOR_WSL parameter is optional.

If you specify Y, the WORKLIST_NAME parameter is required and is used for the JCL member name. Optionally, you can use the PDS_FOR_RUN_JCL parameter to specify the name of partitioned data set (PDS). You cannot specify any other ACTION_*xxxx* parameter.

N

Do not generate JCL.

Default:

N

ACTION_GENERATE_WSL

Specifies whether to generate a work statement list (WSL) when **ACTION_COMPARE**=Y. This option is ignored when a value for the **ACTION_COMPARE** parameter is not specified or it is set to N.

Values:

Y

Yes. A work statement list is generated when **ACTION_COMPARE**= Y. Such a work statement list can be run directly by specifying **ACTION_RUN_WSL**=Y.

If you specify Y, you must specify **ACTION_IMPORT_CHANGE**=N.

N

No. A WSL is not generated.

Default:

N

ACTION_IMPORT_CHANGE

Specifies whether a change is imported. If **ACTION_COMPARE = 'N'**, the DDL or delta change files that are defined by the files that begin with IMCHG (for example, IMCHG001, IMCHG002, and so on) are imported as a new change. If **ACTION_COMPARE = 'Y'**, the result of the compare is imported as a new change.

Values:

Y

If **ACTION_COMPARE = 'N'**, specifies that the content of files IMCHG001 up through IMCHG999 are imported into a new change. You do not need to define all of the IMCHG* files must be defined. For example, only 2 DDL files or delta change files is imported, you need to define only IMCHG001 and IMCHG002.

If ACTION_COMPARE = 'Y', specifies that the result of the compare is imported as a new change. The contents of the files with names IMCHG001 through IMCHG999 are not imported as a new change.

N

Specifies that no importing of a change is done.

blank

Specifies that this parameter defaults to Y if either of the following is true:

1. **`ACTION_COMPARE = 'N'`**, and the IMCHG001 DD is defined and not empty.
2. **`ACTION_COMPARE = 'Y'`**.

Default:

blank

ACTION_IMPORT_IGNORE

Specifies whether an ignore that is defined by the IMIGNORE DD statement is imported as a new ignore.

Values:

Y

Specifies that the content of the `IMIGNORE DD` statement is imported into a new ignore.

N

Specifies that no importing of an ignore is done.

blank

Specifies that this parameter defaults to Y if the `IMIGNORE DD` statement is defined and not empty.

Default:

blank

ACTION_IMPORT_MASK

Specifies whether a mask that is defined by the `IMMASK DD` is imported as a new mask.

Values:

Y

Specifies that the content of the `IMMASK DD` statement is imported into a new mask.

N

Specifies that no importing of a mask is done.

blank

Specifies that this parameter defaults to Y if the `IMMASK DD` statement is defined and not empty.

Default:

blank

ACTION_RECOVER_CHANGE

Specifies whether to recover the change.

Values:

Y

Specifies to recover the change.

N

Specifies to not recover the change.

Default:

N

ACTION_RUN_CHANGE

Specifies whether to run the change. If a change is also being imported, the change that is run is the newly imported change. Otherwise, the change to be run is identified by the **CHANGE_OWNER** and **CHANGE_NAME** parameters.

Values:

Y

Runs the change.

N

Does not run the change.

Default:

N

ACTION_RUN_WSL

Specifies whether to run a WSL.

Values:

Y

Runs the work statement list. The **PDS_FOR_WSL** parameter must be provided to indicate the PDS where the work statement list resides. The **WORKLIST_NAME** parameter must be provided to specify the PDS member.

N

Does not run the WSL.

Default:

N

ADBTEP2_AC

Specifies whether to use autocheck when a change is run. Certain SQL or utility operations can place an object into check-pending state. If you set the Autocheck (AC) parameter value to YES, run change (ADBTEP2) tracks the statements and processes that can place an object in check-pending. If one of these statements is encountered while running a change, an automatic CHECK DATA is done to remove the check-pending state. For the complete description see "The Batch Restart programs: ADBTEP2 and ADBTEPA" on page 572.

Values:

YES

The automatic check process is performed.

NO

The automatic check process is not performed.

Default:

NO

ADBTEP2_ADVISORYAUTOREBUILD

Specifies whether the product, when a change is run, initiates a REBUILD when an object is in certain rebuild pending states. For the complete description and list of values see "The Batch Restart programs: ADBTEP2 and ADBTEPA" on page 572.

Values:

YES

The product automatically attempts a REBUILD if the object is in the ARBDP state.

However, if the parameter **RUN_REORG_REBUILD** was specified as 'A - All relevant' to generate an explicit REBUILD during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REBUILD that duplicates the explicit REBUILD.

NO

The product does not automatically attempt a REBUILD if the object is in the ARBDP state.

Default:
NO

ADBTEP2_ADVISORYAUTOREORG

Specifies whether the product, when a change is run, initiates a REORG when an object is in certain reorganization-pending states. For the complete description see "The Batch Restart programs: ADBTEP2 and ADBTEPA" on page 572.

Values:

YES

The product automatically attempts a REORG if the object is in AREOR or AREO* state.

However, if the parameter **RUN_REORG_REBUILD** was specified as 'A - All relevant' to generate an explicit REORG during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REORG that duplicates the explicit REORG.

NO

The product does not automatically attempt a REORG if the object is in AREOR or AREO* state.

Default:
NO

ADBTEP2_AUTOREBUILD

Specifies whether the product, when a change is run, initiates a REBUILD when an object is in certain rebuild pending states. For the complete description see "The Batch Restart programs: ADBTEP2 and ADBTEPA" on page 572.

Values:

YES

The product automatically attempts a REBUILD if the object is in one of these states: RBDP, RBDP*, or PSRBD state.

However, if the parameter **RUN_REORG_REBUILD** was specified as 'M - Mandatory' or 'A - All relevant' to generate an explicit REBUILD during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REBUILD that duplicates the explicit REBUILD.

NO

The product does not automatically attempt a REBUILD if the object is in one of these states: RBDP, RBDP*, or PSRBD state.

Default:
YES

ADBTEP2_AUTOREORG

Specifies whether the product, when a change is run, initiates a REORG when an object is in certain reorganization-pending states. For the complete description see "The Batch Restart programs: ADBTEP2 and ADBTEPA" on page 572.

Values:

YES

The product automatically attempts a REORG if the object is in the REORP state.

However, if the parameter **RUN_REORG_REBUILD** was specified as 'M - Mandatory' or 'A - All relevant' to generate an explicit REORG during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REORG that duplicates the explicit REORG.

NO

The product does not automatically attempt a REORG if the object is in the REORP state.

Default:
YES

ADBTEP2_BINDERROR

Specifies how BIND or REBIND errors are handled when running a change. For the complete description see “The Batch Restart programs: ADBTEP2 and ADBTEPA” on page 572.

Values:

MAXE

The failed command is written to the ADBHOLD table. The MAXERROR setting (as specified by the ADBTEP2_MAXE parameter) determines if the processing stops immediately, after *nn* errors, or if the bind error does not stop processing.

SAVE

The failed command is written to the ADBHOLD table. Processing continues.

IGNORE

The failed command is not written to the ADBHOLD table. Processing continues.

Default:

MAXE

ADBTEP2_MAXE

Specifies the maximum number of errors. Specifically, this parameter indicates the number of DSN commands that can fail before the batch job is terminated.

Values:

-1

All errors are ignored. The batch job does not stop for any error.

0

No errors are allowed. The batch job stops on the first error.

1-99

The specified number of errors are ignored. The batch job stops on the next DSN command that fails. For example, if you specify 5, the batch job stops when the sixth DSN command fails.

For more information about MAXE, see “Parameters passed to the ADBTEP2 program” on page 573.

Default:

0

ADBTEP2_PENDINGCHANGESCHECK

Specifies whether a check is made when a change is run to avoid losing any Db2 pending changes as part of a DROP action.

Values:

YES

The DROP is not performed if a Db2 pending change exists.

NO

The DROP is performed without checking for pending changes.

Default:

NO

Related information:

“The Batch Restart programs: ADBTEP2 and ADBTEPA” on page 572

ADBTEP2_RESTART

Specifies the RESTART value that is passed to the ADBTEP2 (adbtepx) program. You can restart a change at the beginning of the change WSL, at the point where the change stopped running in a previous run, or at a particular point in the WSL that you specify.

Values:

Y

Yes. RESTART(YES) is used when ADBTEP2 is called.

N
No. RESTART(NO) is used when ADBTEP2 is called.

F
Force. RESTART(FORCE) is used when ADBTEP2 is called.

U
User. RESTART(*USER_RESTART_NAME*) is used when ADBTEP2 is called. If you specify U, you must specify a value for the **ADBTEP2_USER_RESTART_NAME** parameter.

Default:
Y

ADBTEP2_USER_RESTART_NAME
Specifies the name of the location where you want to restart the ADBTEP2 run of the WSL, irrespective of the previous failure point. If **ADBTEP2_USER_RESTART**=U is specified, this parameter is required.

Values:
A valid restart location. 1 - 128 characters.

Default:
blank

ADBTEP2_RESTART_ENV
Specifies the use of checkpoint record environment variables, if they are present, during the restart and execution of set statements in the WSL prior to the restart point.

Values:

CKPT
Uses the environment variables from the checkpoint record during the restart, if they are present.

INPUT
Runs set statements in the WSL before the restart point.

BOTH
Uses the environment variables from the checkpoint record during the restart, if they are present, and runs set statements in the WSL before the restart point.

Default:
The default value is blank.

ADBTEP2_RESTART_REPORT_ONLY
Specifies whether to actually run the restart job or to only simulate the run.

Values:

Y
Yes

N
No

Default:
blank

ADBTEP2_RESTART_SQLID
Specifies the SQL ID to use at the restart point.

Values:
A valid SQL ID. 1 - 8 characters.

Default:
blank

ADBTEP2_RESTART_SCHEMA
Specifies the current schema special register to use at the restart point. Valid values are 1 - 128 characters. The default value is blank.

Values:
A valid schema name. 1 - 128 characters.

Default:
blank

ADBTEP2_RESTART_SERVER
Specifies the location name of the current server.

Values:
A valid location name. 1 - 16 characters.

Default:
blank

ADBTEP2_RESTART_PATH
Specifies the SQL path that is used when unqualified function names, procedure names, data type names, and module object names in dynamically prepared SQL statements are resolved.

Values:
A valid path name. 1 - 254 characters.

Default:
blank

ADBTEP2_RESTART_PRECISION
Specifies the current precision. Valid values are 1 - 5 characters. The default value is blank.

Values:
A valid precision value. 1 - 5 characters.

Default:
blank

ADBTEP2_RESTART_RULES
Specifies the current rules.

Values:

DB2
SQL statements are run based on Db2 rules.

STD
SQL statements are run based on standard SQL rules.

Default:
The default value is blank.

ADBTEP2_RESTART_DECFLOAT_ROUNDING_MODE
Specifies the system default action that is used to round decimal floating-point values.

Values:

- ROUND_CEILING
- ROUND_DOWN
- ROUND_FLOOR
- ROUND_HALF_DOWN
- ROUND_HALF_EVEN

Default:
blank

ADBTEP2_RESTART_ROUTINE_VERSION
Specifies the CURRENT ROUTINE VERSION special register.

Values:
Valid routine version names. 1 - 64 characters.

Default:
blank

ADBTEP2_RESTART_BUSINESS_TIME
Specifies value of the CURRENT TEMPORAL BUSINESS_TIME special register.

Values:
Valid business time values. 1 - 10 characters.

Default:
blank

ADBTEP2_RESTART_SYSTEM_TIME
Specifies value of the CURRENT TEMPORAL SYSTEM_TIME special register. Valid system times are 1 - 10 characters. The default value is blank.

Values:
Valid system time values. 1 - 10 characters.

Default:
blank

ADBTEP2_RESTART_GETARCHIVE
Specifies the value of GET_ARCHIVE global variable.

Values:

Y
Yes. When a table-reference is an archive-enabled table, the table reference includes rows in the associated archive table.

N
No. When a table-reference is an archive-enabled table, the table reference does not include rows in the associated archive table.

Default:
N

ADBTEP2_RESTART_MOVETOARCHIVE
Specifies the value of MOVE_TO_ARCHIVE global variable.

Values:

Y
Yes. When a row in an archive-enabled table is deleted, a copy of the deleted row is stored in the associated archive table.

N
No. When a row in an archive-enabled table is deleted, a copy of the deleted row is not stored in the associated archive table.

Default:
N

ADBTEP2_RESTART_UNLLOBXML
Specifies whether ADBTEP2 should end with an error or unload the table space when an image copy of a table space is requested and any table in the table space contains LOB or XML columns.

Values:

E
ADBTEP2 ends with an error.

U
The table space is unloaded.

Default:
The default value is blank.

ADBTEP2_RESTART_UNLNOIC

Specifies whether ADBTEP2 should end with an error or unload the table space when an image copy of a table space is requested but an image copy is not found.

Values:

E

ADBTEP2 ends with an error.

U

The table space is unloaded.

Default:

The default value is blank.

ADBTEP2_RETRY_DEPRECATED_OBJ

Specifies whether ADBTEP2 is to retry an SQL statement at a lower Db2 function level to create a deprecated object.

This parameter applies only if you are running at Db2 12 function level 504 or higher.

Values:

Yes

Retry the statement.

When the current application compatibility level is Db2 12 function level 504 or higher, and an SQL statement returns SQLCODE -20008 (because it is trying to create a deprecated object), ABDTEP2 takes the following actions:

- Sets the APPLCOMPAT value to V12R1M503
- Retries the statement
- Changes APPLCOMPAT back to the previous value.

No

Do not retry the statement.

Default:

Yes

ADBTEP2_SPANNED

Specifies the SPANNED value that is passed to the ADBTEP2 (adbtepx) program. For the complete description and list of values, see "The Batch Restart programs: ADBTEP2 and ADBTEPA" on page 572.

Values:

Yes

Adds a SPANNED keyword to the UNLOAD statement when the table that is being unloaded has an XML or LOB column.

No

Spanned keyword is not added to Unload statement.

Default:

No

ADBTEP2_STOGROUP_AUTO_REORG_REBUILD

Specifies whether the product, when a change is run, initiates a REORG or REBUILD for the table space or index to implement the effect of altering STOGROUP attribute. For the complete description and list of values see Using the Batch Restart programs: ADBTEP2 and ADBTEPA.

Values:

YES

The product automatically attempts a REORG or REBUILD for the table space or index after SQL statement ALTER STOGROUP is executed. However, if the parameter RUN_REORG_REBUILD was specified as 'A - All relevant' to generate an explicit REORG or

REBUILD during the change flow, then the value NO is passed to ADBTEP2. The NO value prevents an automatic REORG or REBUILD that duplicates the explicit REORG or REBUILD.

NO

The product does not automatically attempt a REORG or REBUILD for the table space or index after SQL statement ALTER STOGROUP is executed.

Default:

NO

ADBTEP2_TIMEOUT_RETRIES

Specifies the maximum number of times that ADBTEP2 is to retry either executing the statement or restarting from the last checkpoint when one of the following conditions occurs:

- For a timeout condition with SQLCODE -913 and reason code x'00C9008E' or a resource unavailable condition (SQLCODE -904 and reason code x'00C200EA'), ADBTEP2 retries executing the statement.
- For a timeout condition with SQLCODE -911 and reason code x'00C9008E', ADBTEP2 tries restarting from the last checkpoint.

Values:

An integer value from 0 to 99. A value of 0 means that ADBTEP2 is not to attempt any retries.

Default:

0

ADBTEP2_TIMEOUT_WAIT_TIME

Specifies the duration, in seconds, between retries by ADBTEP2.

Values:

An integer value from 1 to 3600.

Default:

120

ADBTEP2_TSACCESS

Specifies whether to preserve the access state of the table spaces being accessed. If you set the ADBTEP2_TSACCESS parameter value to YES, run change (ADBTEP2) tracks the statements and processes that can place a table space in a different access state and preserves the state the table space is in.

Values:

YES

The access state of the table space is preserved.

NO

The access state of the table space is not preserved.

Default:

NO

ADMIN_DATASET_BUFNO

Specifies the BUFNO attribute of the TSO ALLOCATE statement for the Db2 Admin Tool data set. The BUFNO attribute is for the number of buffers to be assigned for data control blocks. For more information, see the ADMIN_DATASET_TYPE parameter.

Values:

An integer value 1-255, blank

blank

The BUFNO attribute is not specified for the ALLOCATE statement.

Default:

blank

ADMIN_DATASET_DATACLAS

Specifies the DATACLAS attribute of the TSO ALLOCATE statement for the Db2 Admin Tool data set. The DATACLAS attribute is for the data class name. For more information, see the ADMIN_DATASET_TYPE parameter.

Values:

A valid data class name

Db2 Admin Tool does not validate this value. If an invalid value is specified, an error message is generated from TSO when the allocate of the data set is attempted. Db2 Admin Tool then sets the RECFM, LRECL, and BLKSIZE attributes by specifying these attributes on the ALLOCATE statement. By default, Db2 Admin Tool specifies the space attributes on the allocate statement but you can omit the space attributes from the ALLOCATE statement by specifying ADMIN_DATASET_SPACE_PRIQTY = '<NONE>' for the Db2 Admin Tool.

blank

The DATACLAS attribute is not specified for the ALLOCATE statement.

Default:

blank

ADMIN_DATASET_DEVICE_UNIT

Specifies the device unit for the Db2 Admin Tool data set. For more information, see the ADMIN_DATASET_TYPE parameter.

Values:

A valid device unit, <NONE>

<NONE>

Specifies that the UNIT clause is omitted from the ALLOCATE statement.

Default:

SPACE_UNIT_NAME (See “SPACE_UNIT_NAME” on page 721.)

ADMIN_DATASET_DIR

Specifies the DIR attribute of the TSO ALLOCATE statement for the Db2 Admin Tool data set. The DIR attribute is for the number of directory blocks. For more information, see the ADMIN_DATASET_TYPE parameter. This parameter is only used for the following types of Db2 Admin Tool data sets: IFF, DELTA, DDL SRCVF, TGTVF, MTC.

If the SPACE(priqty,secqty) clause is omitted, then no default value is specified.

Values:

An integer greater than zero, blank

blank

If the SPACE(priqty,secqty) clause is not to be omitted, specifies that the following default values are used for the Db2 Admin Tool data set type that is in effect:

- IFF: 60. A user specified value for directory blocks that you specify only if the Db2 Admin Tool default is insufficient for the change that is being analyzed.
- DELTA: 60
- DDL: 60
- SRCVF: 60
- TGTVF: 60
- MTC: 60

Default:

blank

ADMIN_DATASET_DSN

Specifies the data set name for the Db2 Admin Tool data set. For more information, see the ADMIN_DATASET_TYPE parameter.

Values:

A valid data set name.
The data set name can be 1 to 46 characters or blank.

blank
Specifies that the following default values are to be used for the indicated Db2 Admin Tool data set type that is in effect:

- CHG: &SSID..&CHGTAG..CHG
- DDL: &SSID..&CHGTAG..T&TIME..DDL
- DELTA: D&DATE..T&TIME..DELTA
- IFF: &SSID..&CHGTAG..IFF
- MTC: &SSID..D&DATE..T&TIME..MTC
- SRCVF: OC.D&DATE..T&TIME..SRCVF
- TGTVF: OC.D&DATE..T&TIME..TGTVF

Default:
blank

ADMIN_DATASET_DSNTYPE
Specifies the DSNTYPE attribute of the TSO ALLOCATE statement for the Db2 Admin Tool data set. The DSNTYPE attribute is for the type of data set. For more information, see the ADMIN_DATASET_TYPE parameter. This parameter is only used for the following types of Db2 Admin Tool data sets: IFF, DELTA, DDL SRCVF, TGTVF, MTC.

Values:
LIBRARY, PDS, blank

blank
For data set type IFF, the default is PDS. Otherwise the DSNTYPE attribute is not added to the ALLOCATE statement.

Default:
blank

ADMIN_DATASET_EXPDT
Specifies the EXPDT attribute of the TSO ALLOCATE statement for the Db2 Admin Tool data set. The EXPDT attribute is an expiration date. For more information, see the ADMIN_DATASET_TYPE parameter. This parameter is mutually exclusive with the **ADMIN_DATASET_RETPD** parameter.

Values:
A valid expiration date as defined for the EXPDT attribute for the TSO ALLOCATE statement, blank

blank
The EXPDT attribute is not specified for the ALLOCATE statement.

Default:
blank

ADMIN_DATASET_MAXVOL
Specifies the MAXVOL attribute of the TSO ALLOCATE statement for the Db2 Admin Tool data set. For more information, see the ADMIN_DATASET_TYPE parameter.

Values:

A valid maxvol value as defined by the TSO ALLOCATE statement
Db2 Admin Tool does not validate this value. If an invalid value is specified, an error message is generated from TSO when the allocate of the data set is attempted.

blank
The MAXVOL attribute is not specified for the ALLOCATE statement.

Default:
blank

ADMIN_DATASET_MGMTCLAS

Specifies the MGMTCLAS attribute of the TSO ALLOCATE statement for the Db2 Admin Tool data set. The MGMTCLAS attribute is for the management class name. For more information, see the ADMIN_DATASET_TYPE parameter.

Values:

A valid management class name

Db2 Admin Tool does not validate this value. If an invalid value is specified, an error message is generated from TSO when the allocate of the data set is attempted.

blank

The MGMTCLAS attribute is not specified for the ALLOCATE statement.

Default:

blank

ADMIN_DATASET_RETPD

Specifies the RETPD attribute of the TSO ALLOCATE statement for the Db2 Admin Tool data set. The RETPD attribute is a retention period specified in number of days. For more information, see the ADMIN_DATASET_TYPE parameter. This parameter is mutually exclusive with the **ADMIN_DATASET_EXPDT** parameter.

Values:

An integer value representing the number of days, blank

blank

The RETPD attribute is not specified for the ALLOCATE statement.

Default:

blank

ADMIN_DATASET_SPACE_PRIQTY

Specifies the primary quantity for the Db2 Admin Tool data set. For more information, see the ADMIN_DATASET_TYPE parameter.

Values:

A valid PRIQTY value, <NONE>, blank

<NONE>

Specifies that the SPACE(priqty,secqty), unit of space clauses, and space directory attributes be omitted from the ALLOCATE statement.

blank

Specifies that the following default values are to be used for the indicated Db2 Admin Tool data set type that is in effect:

- CHG: 10
- DDL: 10
- DELTA: 10
- IFF: 2
- MTC: 10
- SRCVF: 10
- TGTVF: 10

Default:

blank

ADMIN_DATASET_SPACE_SECQTY

Specifies the secondary quantity for the Db2 Admin Tool data set. For more information, see the ADMIN_DATASET_TYPE parameter.

If the SPACE(priqty,secqty) clause is omitted, then no default value is specified.

Values:

A valid SECQTY value, blank

blank

If the SPACE(priqty,secqty) clause is not to be omitted, the following default values are used for the Db2 Admin Tool data set type that is in effect:

- CHG: 10
- DDL: 10
- DELTA: 10
- IFF: 2
- MTC: 10
- SRCVF: 10
- TGTVF: 10

Default:

blank

ADMIN_DATASET_SPACE_TYPE

Specifies the space unit type for the Db2 Admin Tool data set.

If the SPACE(priqty,secqty) clause is omitted, no default value is specified.

Values:

CYL

Specifies that the space unit type is cylinders.

TRK

Specifies that the space unit type is tracks.

blank

Specifies that the following default values are used for the Db2 Admin Tool data set type that is in effect:

- CHG: CYL
- DDL: CYL
- DELTA: CYL
- IFF: CYL
- MTC: CYL
- SRCVF: CYL
- TGTVF: CYL

Default:

blank

Related information:

ADMIN_DATASET_TYPE

ADMIN_DATASET_STORCLAS

Specifies the STORCLAS attribute of the TSO ALLOCATE statement for the Db2 Admin Tool data set. The STORCLAS attribute is for the storage class name. For more information, see the ADMIN_DATASET_TYPE parameter.

Values:

A valid storage class name

Db2 Admin Tool does not validate this value. If an invalid value is specified, an error message is generated from TSO when the allocate of the data set is attempted.

blank

The STORCLAS attribute is not specified for the ALLOCATE statement.

Default:
blank

ADMIN_DATASET_TYPE

Specifies the type of data set to which the subsequent Db2 Admin Tool data set type template parameters apply.

You can specify multiple data set types. Separate each type with a semicolon. All data set template parameters that are specified before the next semicolon in the parameter list apply to the type of data set specified. For more information about the syntax of specifying the data set type parameters, see "Db2 Admin Tool data set templates" on page 663.

Values:

CHG

The CHG data set.

Recommendation: When generating a recover change, use CHG_RO or CHG_RE. Specifying CHG when generating a recover change affects the CHG data set for only the change itself; it does not affect the CHG data set for the recover change. If CHG_RO or CHG_RE is not specified, the default recover change CHG data set is created.

CHG_RO

The CHG data set for a recover to original data change. A recover to original data change is specified by the following two options:

- GENERATE_RECOVER_CHANGE = 'Y'
- DATA_TO_RECOVER = 'O' or DATA_TO_RECOVER = 'B'

CHG_RE

The CHG data set for a recover to existing data change. A recover to existing data change is specified by the following two options:

- GENERATE_RECOVER_CHANGE = 'Y'
- DATA_TO_RECOVER = 'E' or DATA_TO_RECOVER = 'B'

DDL

The DDL Admin data set. This data set is the output data set when generating DDL from a base version.

DELTA

The compare delta change file.

IFF

The IFF PDS Db2 Admin Tool data set.

Recommendation: When generating a recover change, use IFF_RO or IFF_RE. Specifying IFF when generating a recover change affects the IFF data set for only the change itself; it does not affect the IFF data set for the recover change. If IFF_RO or IFF_RE is not specified, the default recover change IFF data set is created.

IFF_RO

The IFF PDS Db2 Admin Tool data set for a recover to original data change. A recover to original data change is specified by the following two options:

- GENERATE_RECOVER_CHANGE = 'Y'
- DATA_TO_RECOVER = 'O' or DATA_TO_RECOVER = 'B'

IFF_RE

The IFF PDS Db2 Admin Tool data set for a recover to existing data change. A recover to existing data change is specified by the following two options:

- GENERATE_RECOVER_CHANGE = 'Y'
- DATA_TO_RECOVER = 'E' or DATA_TO_RECOVER = 'B'

MTC
The multi-target change file.

SRCVF
A compare source version work file.

TGTVF
A compare target version work file.

blank
The Db2 Admin Tool data set template parameters are ignored until a supported value for ADMIN_DATASET_TYPE is specified.

Default:
blank

ADMIN_DATASET_VOLUME
Specifies the VOLUME attribute of the TSO ALLOCATE statement for the Db2 Admin Tool data set.

Values:
One or more serial numbers separated by a comma, blank

blank
The VOLUME attribute is not specified for the ALLOCATE statement.

Default:
blank

Related information:

ADMIN_DATASET_TYPE

ALLOW_IMPLICIT_DROP_OF_EXCLUDED_OBJECTS
Specifies whether excluded objects can be dropped implicitly.

Values:

YES
Excluded objects can be dropped implicitly.

NO
Excluded objects cannot be dropped implicitly.

Default:
NO

ALLOW_PBR2_TO_PBR_CHANGES
Specifies whether PBR2 table space objects can be reverted to PBR.

Values:

Y
Revert PBR2 table space objects back to PBR.

N
Do not revert PBR2 table space objects back to PBR.

Default:
N

ALLOW_ROTATE_PARTS
Specifies whether to generate the rotate partition or alter partition statement when the condition for a rotate is met.

Values:

Y
Generate the rotate partition statement. Data from the rotating partitions is unloaded before the rotate takes place. You can either reload the data or discard it.

N

Generate the alter partition statement. Data from the rotating partitions is reloaded into the table. Logical and physical partitions are preserved.

Default:

Y

AUTH_SWITCH_SECADM

Specifies the SECADM authority to use when auth-switching is enabled. The SECADM authority is used to manage all security-related tasks. This parameter applies only if the facility has been enabled for the subsystem as part of the customization process, and applies only when Db2 Admin Tool is connected to Db2 10 or later.

Values:

An SQLID with SECADM authority

Specify a SECADM authority to manage all security-related tasks.

Default:

blank

AUTH_SWITCH_USERID

Specifies the auth-switch ID to use when auth-switching is enabled. This parameter applies only when the facility has been enabled for the subsystem as part of customization process.

Values:

An SQLID

The ID to connect as when auth-switching.

<NONE>

Avoids producing auth-switch work-statement lists (WSL).

<SQLID>

Enables the SQLID authorization switching feature.

blank

Produces auth-switch WSL, with the ID portion of the WSL as comments.

Default:

<NONE>

AUTH_SWITCHING_ENABLED

Specifies whether auth-switching is enabled.

Values:

Y

Auth-switching is used if an auth-switch ID is specified.

N

Auth-switching is used.

Default:

N

BASE_VERSION_NAME

Specifies the name of the base version to perform the action on. If a base version is being saved or generated, this parameter specifies the name for the new base version if the other base version name parameters are blank. If a base version is not being saved or generated, the value of this parameter must identify the name of an existing base version.

The base version parameter hierarchy is as follows:

- BASE_VERSION_OWNER
  - NEW_BASE_VERSION_OWNER
    - BASE_VERSION_OWNER_BEFORE_RUN
    - BASE_VERSION_OWNER_AFTER_RUN

- BASE_VERSION_NAME
  - NEW_BASE_VERSION_NAME
    - BASE_VERSION_NAME_BEFORE_RUN
    - BASE_VERSION_NAME_AFTER_RUN

Values:
A valid 1- to 128-character version name.

Default:
AUTO:&CURTS.

BASE_VERSION_NAME_AFTER_RUN
Specifies the name for a new base version that is created after a change is implemented.

The base version parameter hierarchy is as follows:

- BASE_VERSION_OWNER
  - NEW_BASE_VERSION_OWNER
    - BASE_VERSION_OWNER_BEFORE_RUN
    - BASE_VERSION_OWNER_AFTER_RUN
- BASE_VERSION_NAME
  - NEW_BASE_VERSION_NAME
    - BASE_VERSION_NAME_BEFORE_RUN
    - BASE_VERSION_NAME_AFTER_RUN

Values:
Valid version name; 1 to 128 characters

Default:
NEW_BASE_VERSION_NAME

BASE_VERSION_NAME_BEFORE_RUN
Specifies the name for a new base version that is created before a change is implemented.

Base version parameter hierarchy:

- BASE_VERSION_OWNER
  - NEW_BASE_VERSION_OWNER
    - BASE_VERSION_OWNER_BEFORE_RUN
    - BASE_VERSION_OWNER_AFTER_RUN
- BASE_VERSION_NAME
  - NEW_BASE_VERSION_NAME
    - BASE_VERSION_NAME_BEFORE_RUN
    - BASE_VERSION_NAME_AFTER_RUN

Values:
Valid version name; 1 to 128 characters

Default:
NEW_BASE_VERSION_NAME

BASE_VERSION_OWNER
Specifies the owner of the base version to perform the action on. If a base version is being saved or generated, this parameter specifies the owner for the new base version if the other base version owner parameters are blank. If a base version is not being saved or generated, the value of this parameter must identify the owner of an existing base version.

Base version parameter hierarchy:

- BASE_VERSION_OWNER
  - NEW_BASE_VERSION_OWNER
    - BASE_VERSION_OWNER_BEFORE_RUN
    - BASE_VERSION_OWNER_AFTER_RUN
- BASE_VERSION_NAME
  - NEW_BASE_VERSION_NAME
    - BASE_VERSION_NAME_BEFORE_RUN
    - BASE_VERSION_NAME_AFTER_RUN

Values:
Valid version name; 1 to 128 characters

Default:
&CURSQLID.

BASE_VERSION_OWNER_AFTER

Specifies the owner for a new base version that is created after a change is implemented.

Base version parameter hierarchy:

- BASE_VERSION_OWNER
  - NEW_BASE_VERSION_OWNER
    - BASE_VERSION_OWNER_BEFORE_RUN
    - BASE_VERSION_OWNER_AFTER_RUN
- BASE_VERSION_NAME
  - NEW_BASE_VERSION_NAME
    - BASE_VERSION_NAME_BEFORE_RUN
    - BASE_VERSION_NAME_AFTER_RUN

Values:
Valid version name; 1 to 128 characters

Default:
NEW_BASE_VERSION_OWNER

BASE_VERSION_OWNER_BEFORE_RUN

Specifies the owner for a new base version that is created before a change is implemented.

Base version parameter hierarchy:

- BASE_VERSION_OWNER
  - NEW_BASE_VERSION_OWNER
    - BASE_VERSION_OWNER_BEFORE_RUN
    - BASE_VERSION_OWNER_AFTER_RUN
- BASE_VERSION_NAME
  - NEW_BASE_VERSION_NAME
    - BASE_VERSION_NAME_BEFORE_RUN
    - BASE_VERSION_NAME_AFTER_RUN

Values:
Valid version name; 1 to 128 characters

Default:
NEW_BASE_VERSION_OWNER

BASE_VERSION_SCOPE_NAME

Specifies the default name of an existing version scope to use when generating a new base version using the USER method.

Base version scope parameter hierarchy:

- BASE_VERSION_SCOPE_OWNER
  - BASE_VERSION_SCOPE_OWNER_BEFORE_RUN
  - BASE_VERSION_SCOPE_OWNER_AFTER_RUN
- BASE_VERSION_SCOPE_NAME
  - BASE_VERSION_SCOPE_NAME_BEFORE_RUN
  - BASE_VERSION_SCOPE_NAME_AFTER_RUN

Values:

Valid version name; 1 to 128 characters, blank

Default:

blank

BASE_VERSION_SCOPE_NAME_AFTER_RUN

Specifies the name of an existing version scope to use when generating a new base version after a change is implemented. This applies only if the new base version is created using the USER method.

Base version scope parameter hierarchy:

- BASE_VERSION_SCOPE_OWNER
  - BASE_VERSION_SCOPE_OWNER_BEFORE_RUN
  - BASE_VERSION_SCOPE_OWNER_AFTER_RUN
- BASE_VERSION_SCOPE_NAME
  - BASE_VERSION_SCOPE_NAME_BEFORE_RUN
  - BASE_VERSION_SCOPE_NAME_AFTER_RUN

Values:

Valid version name; 1 to 128 characters

Default:

BASE_VERSION_SCOPE_NAME

BASE_VERSION_SCOPE_NAME_BEFORE_RUN

Specifies the name of an existing version scope to use when generating a new base version before a change is implemented. This applies only if the new base version is created using the USER method.

Base version scope parameter hierarchy:

- BASE_VERSION_SCOPE_OWNER
  - BASE_VERSION_SCOPE_OWNER_BEFORE_RUN
  - BASE_VERSION_SCOPE_OWNER_AFTER_RUN
- BASE_VERSION_SCOPE_NAME
  - BASE_VERSION_SCOPE_NAME_BEFORE_RUN
  - BASE_VERSION_SCOPE_NAME_AFTER_RUN

Values:

Valid version name; 1 to 128 characters

Default:

BASE_VERSION_SCOPE_NAME

BASE_VERSION_SCOPE_OWNER

Specifies the default owner of an existing version scope to use when generating a new base version using the USER method.

Base version scope parameter hierarchy:

- BASE_VERSION_SCOPE_OWNER
  - BASE_VERSION_SCOPE_OWNER_BEFORE_RUN
  - BASE_VERSION_SCOPE_OWNER_AFTER_RUN
- BASE_VERSION_SCOPE_NAME
  - BASE_VERSION_SCOPE_NAME_BEFORE_RUN
  - BASE_VERSION_SCOPE_NAME_AFTER_RUN

Values:
Valid version scope owner; 1 to 128 characters

Default:
&CURSQLID.

BASE_VERSION_SCOPE_OWNER_AFTER_RUN
Specifies the owner of an existing version scope to use when generating a new base version after a change is implemented. This applies only if the new base version is created using the USER method

Base version scope parameter hierarchy:

- BASE_VERSION_SCOPE_OWNER
  - BASE_VERSION_SCOPE_OWNER_BEFORE_RUN
  - BASE_VERSION_SCOPE_OWNER_AFTER_RUN
- BASE_VERSION_SCOPE_NAME
  - BASE_VERSION_SCOPE_NAME_BEFORE_RUN
  - BASE_VERSION_SCOPE_NAME_AFTER_RUN

Values:
Valid version scope owner; 1 to 128 characters

Default:
BASE_VERSION_SCOPE_OWNER

BASE_VERSION_SCOPE_OWNER_BEFORE_RUN
Specifies the owner of an existing version scope to use when generating a new base version before a change is implemented. This applies only if the new base version is created using the USER method

Base version scope parameter hierarchy:

- BASE_VERSION_SCOPE_OWNER
  - BASE_VERSION_SCOPE_OWNER_BEFORE_RUN
  - BASE_VERSION_SCOPE_OWNER_AFTER_RUN
- BASE_VERSION_SCOPE_NAME
  - BASE_VERSION_SCOPE_NAME_BEFORE_RUN
  - BASE_VERSION_SCOPE_NAME_AFTER_RUN

Values:
Valid version scope owner; 1 to 128 characters

Default:
BASE_VERSION_SCOPE_OWNER

CHANGE_COMMENT
Specifies the comment for a new change.

Values:
1 to 128 characters

Default:
blank

CHANGE_NAME

Specifies the name of the change to perform the action on. If a change is being imported, this parameter specifies the name for the new change if the value for the **NEW_CHANGE_NAME** parameter is blank. If a change is not being imported, the value for this parameter must identify the name of an existing change.

Values:

Valid change name; 1 to 128 characters. If CHGTAG_TYPE ='NAME', the change name must be 1 to 8 characters due to data set naming restrictions.

Default:

AUTO:&CURTS.

CHANGE_OWNER

Specifies the owner of the change to perform the action on. If a change is being imported, this parameter specifies the owner for the new change if the value for the **NEW_CHANGE_OWNER** parameter is blank. If a change is not being imported, the value for this parameter must identify the owner of an existing change.

Values:

Valid change owner; 1 to 128 characters

Default:

&CURSQLID.

CHGTAG_TYPE

Specifies the type of values that the Db2 Admin Tool &CHGTAG. symbol variable resolves to. Refer to the product-defined symbol variables information and the definition of the Db2 Admin Tool &CHGTAG. symbol variable for details.

Values:

ID

The &CHGTAG. symbol variable value is based on the Db2 Admin Tool generated change ID.

NAME

The &CHGTAG. symbol variable value is based on the user specified change name.

OWNER

The &CHGTAG. symbol variable value is based on the user specified change owner.

Default:

ID

COMPARE_IGNORE_CHANGES_NAME

Specifies the name of an existing Ignore Changes Specification that is stored in the Change Management database. The **COMPARE_IGNORE_CHANGES_OWNER** and **COMPARE_IGNORE_CHANGES_NAME** parameters uniquely identify an Ignore Changes Specification to be used during the compare process.

Values:

A valid ignore changes name, blank

A valid ignore changes name; 1 to 128 characters

Specify a 1- to 128-character Ignore Changes Specification is used during the compare process.

blank

Ignore changes is not used during the compare process.

Default:

blank

COMPARE_IGNORE_CHANGES_OWNER

Specifies the owner of an existing Ignore Changes Specification that is stored in the Change Management database. The **COMPARE_IGNORE_CHANGES_OWNER** and **COMPARE_IGNORE_CHANGES_NAME** parameters uniquely identify an Ignore Changes Specification to be used during the compare process.

Values:

Specify a valid 1- to 128-character ignore changes owner.

Default:

&CURSQLID.

COMPARE_IGNORE_FIELDS_DSN

Specifies the name of a data set that contains the ignore fields to be used during the compare. The **PREFIX_FOR_DATA_SETS** parameter is used to qualify the data set name if the specified data set name is not fully qualified. If the compare ignore fields file IGNORES DD is pre-allocated and this parameter is specified, the ignore fields specified by this parameter are used instead of the pre-allocated compare ignore fields file. The **COMPARE_IGNORE_FIELDS_DSN** and **COMPARE_IGNORE_FIELDS_NAME** parameters are mutually exclusive.

Values:

A valid data set name

The data set must contain ignore fields and be a fixed block sequential data set or a member of a partitioned data set with a record length of 80 (RECFM=F*x*, LRECL=80). The input must be in columns 1-72 of the data set.

Syntax:

```
objecttype: field1,field2, .... ,fieldn
```

where **objecttype** is the Db2 catalog table name and fieldn : is the Db2 catalog column to be ignored

Examples:

```
SYSDATABASE: BPOOL
SYSDATABASE: INDEXBP,STGROUP
SYSTABLESPACE: BPOOL
SYSTABLEPART: PQTY,SQTY,STORNAME,VCATNAME
SYSINDEXES: INDEXSPACE
SYSINDEXPART: PQTY,SQTY,STORNAME,VCATNAME
```

Ignore fields are applied to both the target and the source objects before the definitions are compared.

Related information:

Translation masks and ignore fields (IBM Db2 Object Comparison Tool for z/OS 12.1.0)

Default:

blank

COMPARE_IGNORE_FIELDS_NAME

Specifies the name of an existing Ignore Fields Specification that is stored in the Change Management database. The **COMPARE_IGNORE_FIELDS_OWNER** and **COMPARE_IGNORE_FIELDS_NAME** parameters uniquely identify the Ignore Fields Specification to be used during the compare process. If the compare ignore fields file IGNORES DD is pre-allocated and this parameter is specified, the ignore fields that are specified by this parameter are used instead of the pre-allocated compare ignore fields file. The **COMPARE_IGNORE_FIELDS_DSN** and **COMPARE_IGNORE_FIELDS_NAME** parameters are mutually exclusive.

Values:

A valid ignore fields name, blank

A valid ignore fields name

Specify a 1- to 128-character Ignore Fields name. The specified Ignore Fields Specification is used during the compare process.

Default:

blank

COMPARE_IGNORE_FIELDS_OWNER

Specifies the owner of an existing Ignore Fields Specification that is stored in the Change Management database. The **COMPARE_IGNORE_FIELDS_OWNER** and **COMPARE_IGNORE_FIELDS_NAME** parameters uniquely identify the Ignore Fields Specification to be used during the compare process.

Values:

Specify a 1- to 128-character Ignore Fields owner.

Default:

&CURSQLID.

COMPARE_MASK_DSN

Specifies the name of a data set that contains the masks to be used for the compare. The **PREFIX_FOR_DATA_SETS** parameter is used to qualify the data set name if the specified data set name is not fully qualified. If the compare masks file MASKS DD is pre-allocated and this parameter is specified, the masks that are specified by this parameter are used instead of the pre-allocated compare masks file. The **COMPARE_MASK_DSN** and **COMPARE_MASK_NAME** parameters are mutually exclusive.

Values:

A valid data set name

The data set must contain masks and must be a fixed block sequential data set or a member of a partitioned data set with a record length of 80 (RECFM=F*x*, LRECL=80). The input must be in columns 1-72 of the data set.

Here are some mask definition examples:

```
NAME: ABC*, DEF*
NAME: HLQ*D*, NEW**
OWNER: SYSIBM,MYCAT
```

Masks are applied to the source objects before they are compared with the target. You can define as many masks as you want; however, defining many masks will degrade the performance of compare. The first left hand mask that matches are used and the name is translated to the right hand value. If no match is found it is not translated, but still participate in the compare. Using the above masks a source database with the name 'HLQ47D9' is translated to 'NEW479' before it is compared with the target databases.

Related information:

Translation masks and ignore fields (IBM Db2 Object Comparison Tool for z/OS 12.1.0)

Default:

blank

COMPARE_MASK_NAME

Specifies the name of an existing mask entry that is stored in the Change Management database that is to be used for the compare. The **COMPARE_MASK_OWNER** and **COMPARE_MASK_NAME** parameters uniquely identify the mask entry to be used during the compare process. If the compare masks file MASKS DD is pre-allocated and this parameter is specified, the masks that are specified by this parameter are used instead of the pre-allocated compare masks file. The **COMPARE_MASK_DSN** and **COMPARE_MASK_NAME** parameters are mutually exclusive.

Values:

Specify a valid 1- to 128-character mask name.

Default:

blank

COMPARE_MASK_OWNER

Specifies the owner of an existing mask entry that is stored in the Change Management database that is to be used for the compare. The **COMPARE_MASK_OWNER** and **COMPARE_MASK_NAME** parameters uniquely identify the mask entry to be used during the compare process.

Values:

Specify a valid 1- to 128-character mask owner.

Default:

&CURSQLID.

COMPARE_RESULTS_COMMENT

Specifies a comment for the saved compare result. You can use this comment parameter to describe the nature of the compare run. This comment is stored with the saved compare result.

Values:

Specify a 1- to 128-character comment or leave this parameter blank.

Default:

blank

COMPARE_RESULTS_ELIGIBLE_FOR_AUTO_DELETE

Specifies when the saved compare result is eligible for deletion by the auto-delete process of Db2 Admin Tool.

Values:

Number of days until eligible for auto-delete

Specify a number in the range 1-9999.

blank

No auto-deletion will take place.

Default:

blank

COMPARE_RESULTS_NAME

Specifies the name for the compare result that is stored in the Change Management database. The **COMPARE_RESULTS_OWNER** and **COMPARE_RESULTS_NAME** together uniquely identify the saved compare result. You can manage the saved compare result using the "MR - Manage saved compare results" dialogs, which you access from the Object Comparison Tool main menu.

Values:

Name for the compare results.

Specify a valid 1- to 128-character compare results name.

Default:

AUTO:&CURTS.

COMPARE_RESULTS_OWNER

Specifies the owner for the compare result that is stored in the Change Management database. The **COMPARE_RESULTS_OWNER** and **COMPARE_RESULTS_NAME** together uniquely identify the saved compare result. You can manage the saved compare result using the "MR - Manage saved compare results" dialogs, which you access from the Object Comparison Tool main menu.

Values:

Owner of the compare result

A valid 1- to 128 character name of the compare results owner.

Default:

&CURSQLID.

CONTENT_OF_APPLY_JOBS

Specifies whether to generate changes only to database objects and to not generate unloads, loads or other utilities, except REBIND.

Values:

A

Generate all jobs and processes to reload data.

D

Generate only SQL.

Restriction: You must set the **CONTENT_OF_APPLY_JOBS** parameter to A if the **GENERATE_RECOVER_CHANGE** parameter is set to Yes.

Default:
A

DATA_TO_RECOVER
Specifies the type of data that the recover change recovers.

Values:

O
Recover using the original data. The *original data* is the data that is unloaded when the original change is run. If you use the original data during a recovery operation, consider whether related tables that were not affected by the recover must also be restored to the same point to avoid inconsistencies. This option applies only to tables that were unloaded in the original change.

E
Recover using the existing data. The *existing data* is the data that exists in the table just before the original change is recovered. If a table is dropped without being re-created in the original change, no data is loaded after the table is created in the recover change.

B
Set up recover with both the existing data and the original data, so that you will have the choice to recover with either type of data. (At the time of recovery, you can specify RECOVER_OPTION=’O’ or RECOVER_OPTION=’E’).

N
Do not use any data to do recover. This option allows for a recovery with DDL only.

Default:
E

DB2_SORT_LINKLIB
Specifies the Db2 Sort SCNKLINK library. This data set must be APF-authorized. This parameter is only valid when parameter Unload_method = 'H' for Db2 High Performance Unload is specified too.

Values:
Valid library names. 1 - 44 characters.

Default:
blank

DB2_SORT_LPALIB
Specifies the Db2 Sort SCNKLPA library. This data set must be APF-authorized. This parameter is only valid when parameter Unload_method = 'H' for Db2 High Performance Unload is specified too.

Values:
Valid library names. 1 - 44 characters.

Default:
blank

DEFAULT_SPACE_PRIQTY
Specifies the default primary space allocation. The default space allocation values are used to allocate, copy, and unload data sets when RUNSTATS or STOSPACE has not been run.

Values:
Specify a valid PRIQTY value.

Default:
30

DEFAULT_SPACE_SECQTY
Specifies the default secondary space allocation. The default space allocation values are used to allocate, copy, and unload data sets when RUNSTATS or STOSPACE has not been run.

Values:

Specify a valid SECQTY value.

Default:

30

DISABLE_OPTIMIZE_REORG

Controls whether the compare process disables the optimization of REORG statements.

This parameter is useful when multiple REORG operations are performed on the same table space because of a change.

Values:

Y

Disables the optimization of REORG statements.

N

REORG statements are optimized. The minimum number of REORG operations are used to try to avoid multiple REORG operations on the same table space.

Default:

N

DISPLAY_MTC_STATEMENTS

Specifies whether the change statements registered for multi-target changes are displayed when analyzing and running of a multi-target change.

Note: These are the change statements on the central system where multi-target change is registered. On each target system, these statements will have variations based on masks applied.

Values:

Y

Display the multi-target change statements

N

Do not display the multi-target change statements.

Default:

N

DO_NOT_FORCE_SUPPRESS_FOR_AUTO

Specifies whether to force the **Suppress DROP of objects** option when the target is type AUTO.

Values:

YES

Do not force this option.

NO

Force this option.

Default:

N

DO_RUNTIME_ANALYZE

Specifies whether to do a runtime analyze before a change is run. The runtime analyze is a safety check to ensure a change being run is based on the latest Db2 catalog information.

Values:

Y

Perform a run-time analyze. If the product detects that the latest Db2 catalog information is not used but is needed, the run process will fail with an error. The change will need to be analyzed again before it can be run.

N

A runtime analyze is not done before a change is run.

Default:
Y

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.

Values:

YES or Y
If a table on the target has foreign keys and the table also exists on the source, all foreign keys that are not also on the source will be dropped from the target.

NO or N
The drop behavior is determined by the SUPPRESS_DROP_OF_OBJECTS parameter.

Default:
NO

ENABLE_ACCELERATED_TABLES
Specifies whether to enable acceleration for accelerated tables that are automatically recreated and reloaded in DROP/CREATE or ALTER situations.

Values:

Y
Enables accelerated tables.

N
Does not enable accelerated tables.

Default:
Y

If ENABLE_ACCELERATED_TABLES is set to YES, RELOAD_ACCELERATED_TABLES must also be set to YES.

EXISTING_BASE_VERSION_ACTION
Specifies the action to take if a new base version owner and name identify an existing base version.

Values:

REPLACE
The existing base version is replaced with the new base version.

AUTO
The specified base version name is not used. Instead, Db2 Admin Tool uses the product default value for a base version name, such as `AUTO:&CURTS.`. A warning message is issued to notify you of this event.

Default:
AUTO

EXISTING_CHANGE_ACTION
Specifies the action to be taken when a change already exists.

Values:

REPLACE CONDITIONAL
Replace the change if it is not a prerequisite for other changes. If the change is a prerequisite for other changes, an error message is issued. The names of changes that are dependent on the change are displayed and the replace change request is not processed.

REPLACE UNCONDITIONAL
Replace the change even if it is a prerequisite for other changes. The names of changes that are dependent on the change are displayed.

The change status of changes that are dependent on the change is changed to DEFINED, and the changes must be analyzed before being run.

STOP
Do not replace the change.

Default:
STOP

EXISTING_DATA_SET_ACTION
Specifies the action that occurs if a data set with the same name already exists, and if a supported Db2 Admin Tool data sets is needed. The EXISTING_DATA_SET_ACTION parameter does not apply for WSL members when a conversion to ISPF format is performed. The following data set types support the **EXISTING_DATA_SET_ACTION** parameter:

- CHG sequential file
- IFF PDS
- WSL PDS member
- JCL PDS member
- run job input PDS

If a recover change is generated, the existing data set action option also defines the action for Db2 Admin Tool data sets that are associated with the recover change, for example, recover CHG sequential file or recover IFF PDS, and so on.

Values:

CONDITIONAL
If the data set or PDS member already exists, and the data set or PDS member is already associated with the change from a previous CM action, replace the data set or PDS member. If the data set or PDS member already exists, and the data set or PDS member is not already associated with the change, stop processing.

REPLACE
If the data set or PDS member already exists, replace it.

STOP
If the data set or PDS member already exists, stop processing.

Default:
CONDITIONAL

EXISTING_MASK_ACTION
Specifies the action that occurs during import mask if the mask specified by **COMPARE_MASK_OWNER** and **COMPARE_MASK_OWNER** parameters identifies an existing mask entry.

Values:

REPLACE CONDITIONAL
Replace the mask if it is not associated with a registered change that needs the mask for implementation. If the mask is needed by one or more changes for implementation, the names of changes are displayed and the mask is not replaced.

REPLACE CONDITIONAL mode does not cover the scenario in which changes can be associated with a mask but the mask is no longer needed to implement the change. For example, when a change is imported using masking, after the import completes the mask is no longer needed to implement that change. The mask, however, is still associated with the change.

REPLACE UNCONDITIONAL
Replace the mask even if it is associated with a registered change that needs the mask for implementation. The names of changes that need the mask for implementation are displayed and the mask is replaced. Only the changes that need the mask for implementation are reported.

REPLACE UNCONDITIONAL mode does not cover the scenario in which changes can be associated with a mask but the mask is no longer needed to implement the change. For example, when a change is imported using masking, after the import completes the mask is no longer needed to implement that change. In this scenario, REPLACE UNCONDITIONAL does not report the imported change. The mask, however, is still associated with the change.

STOP
Do not replace the mask.

Default:
STOP

GEN_EXCLUDE_NAME
Specifies the name of an Exclude Specification that is stored in the Change Management database. The Exclude Specification is used for the GEN batch job during the CM batch compare.

Values:
A valid 1- to 128-character exclude specification name, blank

Default:
blank

GEN_EXCLUDE_OWNER
Specifies the owner of an Exclude Specification that is stored in the Change Management database. The Exclude Specification is used for the GEN batch job during the CM batch compare.

Values:
A valid 1- to 128-character exclude specification owner.

Default:
&CURSQLID

GENERATE_BASE_VERSION_AFTER_RUN
Specifies whether and how to automatically generate a new base version after a change is implemented. The base version that is generated is associated with the change. Automatically generating a base version after a change is run enables you to keep a record of object definitions after they were changed, and to associate this base version with the change entry.

Values:

AUTO
Db2 Admin Tool automatically determines the objects that are in the base version based on the objects that are being changed.

USER
The objects that are in the base version are defined by a version scope that is specified by the user.

NO
A new base version is not generated after the change is implemented.

Default:
NO

GENERATE_BASE_VERSION_BEFORE_RUN
Specifies whether and how to automatically generate a new base version before a change is implemented. The base version that is generated is associated with the change. Automatically generating a base version after a change is run enables you to keep a record of object definitions after they were changed and to associate this base version with the change entry.

Values:

AUTO
Db2 Admin Tool automatically determines the objects that are in the base version based on the objects that are being changed.

USER
The objects that are in the base version are defined by a version scope that is specified by the user.

NO
A new base version is not generated after the change is implemented.

Default:
NO

GENERATE_JOB_CLASS
Specifies whether to include the **CLASS** parameter on the job card. If you include the **CLASS** parameter on the job card, end the last line of the job card with a comma because Db2 Admin Tool places the **CLASS** parameter on a new line.

Values:

Y
Generate a job class parameter with the value of the **JOB_CLASS** parameter.

N
Do not generate a job class parameter.

Default:
Y

GENERATE_RECOVER_CHANGE
Specifies whether to generate a recover change if the change does not already have a recover change. If the change already has a recover change, the recover change is regenerated.

Values:

Y
A recover change is generated during analyze.

N
If the change does not have a recover change, a recover change is not generated. Otherwise, this parameter is forced to be set to Y and the recover change is regenerated.

Default:
N

GENERATE_TEMPLATES
Specifies whether to generate templates.

Values:

Y
Use the user-defined templates in the ADBTEMPL DD data definition. Refer to Symbol variables in the ADBTEMPL file: Db2 TEMPLATE support for information about using symbol variables to specify Db2 TEMPLATE statements.

N
Use the Db2 Admin Tool default template statements.

Default:
N

GRANT_ORDER
Specifies which prefix the GRANT statement will use.

Values:

C
Generates the GRANT statement with the CREATE prefix.

P
Generates the GRANT statement with the POSTUTIL prefix.

Default:
C

IDENTITY_START_VALUE
Specifies the START value of an IDENTITY column of a table if the table is re-created.

Values:

O
The START value from the Db2 catalog is used.

C
The START value is computed based on the identity attributes of the column.

Default:
O

IGNORE_COMMENT
Specifies the comment for a new ignore.

Values:
Specify a 1- 128-character comment or leave this parameter blank.

Default:
blank

IGNORE_NAME
Specifies the name for an existing ignore or a new ignore, depending on what action the Change Management (CM) batch interface is invoked. If an ignore is being imported and if the value for the **NEW_IGNORE_NAME** parameter is blank, this parameter specifies the name for the new ignore.

Values:
Specify a valid 1- 128-character ignore name.

Default:
AUTO:&CURTS.

IGNORE_OWNER
Specifies the owner for an existing ignore or a new ignore, depending on what action the Change Management (CM) batch interface is invoked. If an ignore is being imported and if the value for the **NEW_IGNORE_OWNER** parameter is blank, this parameter specifies the owner for the new ignore.

Values:
Specify a valid 1- 128-character ignore owner.

Default:
&CURSQLID.

IMPORT_PENDING_CHANGE_ACTION
Specifies the action that occurs if the import data set contains changes to objects that have changes pending from Db2 Admin Tool Change Management.

Values:

P
Make the pending changes a prerequisite for the imported change.

S
Supersede the pending changes and continue importing the change. The pending changes are placed in DEFINED status and will have the superseded change as a prerequisite.

I
Ignore the pending changes and continue importing the change. Analyzed pending changes are left in ANALYZED status and prerequisites are not established.

C
Cancel the import change process.

Default:
P

JOB_CARD_LINE_1
Specifies line 1 of the job card for generated jobs.

Values:
Specify a 1- to 72-character statement.

Default:

//&USERID.D JOB (&SYSUID),'CM BATCH',

JOB_CARD_LINE_2

Specifies line 2 of the job card for generated jobs.

Values:

Specify a 1- to 72-character statement.

Default:

// REGION=0K,NOTIFY=&SYSUID,MSGCLASS=H,MSGLEVEL=(1,1),

JOB_CARD_LINE_3

Specifies line 3 of the job card for generated jobs.

Values:

Specify a 1- to 72-character statement.

Default:

blank

JOB_CARD_LINE_4

Specifies line 4 of the job card for generated jobs.

Values:

Specify a 1- to 72-character statement.

Default:

blank

JOB_CARD_LINE_5

Specifies line 5 of the job card for generated jobs.

Values:

Specify a 1- to 72-character statement.

Default:

blank

JOB_CLASS

Specifies the CLASS parameter value for the job card.

Values:

Specify a valid job class.

Default:

A

JOB_JCLLIB_LINE_1

Specifies line 1 of the JCLLIB statement. The GOCCM JCL procedure must be accessible in the libraries that are defined by the JCLLIB statement in the run job or in the system procedure libraries.

Values:

Specify a 1- to 72-character statement.

Default:

blank

The following example shows how to set this parameter:

```
JOB_JCLLIB_LINE_1 = '//GOCCM JCLLIB ORDER=ADB.DEVCUST.JCLLIB’
```

This example results in the following JCL line in jobs that are generated by Change Management (CM) batch interface:

```
//GOCCM JCLLIB ORDER=ADB.DEVCUST.JCLLIB
```

JOB_JCLLIB_LINE_2

Specifies line 2 of the JCLLIB statement.

Values:
Specify a 1- to 72-character statement.

Default:
blank

JOB_JCLLIB_LINE_3
Specifies line 3 of the JCLLIB statement.

Values:
Specify a 1- to 72-character statement.

Default:
blank

JOB_JCLLIB_LINE_4
Specifies line 4 of the JCLLIB statement.

Values:
Specify a 1- to 72-character statement.

Default:
blank

JOB_PARM_LINE_1
Specifies line 1 of the job parameter area.

Values:
Specify a 1- to 72-character statement.

Default:
blank

The following example shows how to set this parameter:

```
JOB_PARM_LINE_1='S=SYS4A'
```

This example results in the following line in JCL that is generated by Change Management (CM) batch interface:

```
/*JOBPARM S=SYS4A
```

JOB_PARM_LINE_2
Specifies line 2 of the job parameter area.

Values:
Specify a 1- to 72-character statement.

Default:
blank

JOB_PARM_LINE_3
Specifies line 3 of the job parameter area.

Values:
Specify a 1- to 72-character statement.

Default:
blank

JOB_PARM_LINE_4
Specifies line 4 of the job parameter area.

Values:
Specify a 1- to 72-character statement.

Default:
blank

LIST_OPTIONS

Specifies whether ADBMSGS includes the following lists of CM batch parameter values:

- A list of the parameter values that were provided in an invocation override (by using the PARMS DD statement)
- A list of the final parameter values that were used

Values:

Y

These two lists are included. Each list is identified in ADBMSGS by message ADB7957I.

N

These lists are not included.

Default:

N

Related information:

"ADB7957I" on page 1156
"Consolidating messages into a single file" on page 246

LOAD_ACCELERATED_TABLES_DETECT_CHANGES

Specifies the value of detectChanges for the SYSPROC.ACCEL_LOAD_TABLES procedure for accelerated tables.

Values:

DATA

For information about DATA, see the description of *detectChanges* in SYSPROC.ACCEL_LOAD_TABLES (IBM DB2 Analytics Accelerator for z/OS 7.5).

PARTITIONS

For information about PARTITIONS, see the description of *detectChanges* in SYSPROC.ACCEL_LOAD_TABLES (IBM DB2 Analytics Accelerator for z/OS 7.5).

NONE

DetectChanges is not used when calling SYSPROC.ACCEL_LOAD_TABLES.

Default:

DATA

MASK_COMMENT

Specifies the comment for a new mask.

Values:

Specify a 1- 128-character comment or leave this parameter blank.

Default:

blank

MASK_IGNORED_FIELDS

Specifies whether ignores takes precedence for newly added objects when both ignores and masks are specified for a specific attribute.

Values:

YES

Masks are applied to ignored fields.

NO

Ignored fields retain their original values.

Default:

NO

MASK_NAME

Specifies the name for an existing mask or a new mask, depending on what action the Change Management (CM) batch interface is invoked. If a mask is being imported and if the value for the **NEW_MASK_NAME** parameter is blank, this parameter specifies the name for the new mask.

Values:

Specify a 1- 128-character mask name or leave this parameter blank.

Default:

AUTO:&CURTS.

COMPARE_MASK_OWNER

Specifies the owner for an existing mask or a new mask, depending on what action the Change Management (CM) batch interface is invoked. If a mask is imported and if the value for the **NEW_COMPARE_MASK_OWNER** parameter is blank, this parameter specifies the owner for the new mask.

Values:

Specify a 1- 128-character mask owner or leave this parameter blank.

Default:

&CURSQLID.

MAX_ALLOCATION_TO_DASD

Specifies the maximum amount of space that can be allocated to DASD. This parameter applies only to new copy and unload data sets. When the space that is required for an unload or copy data set exceeds this threshold value, the data set is allocated to the tape unit that is specified in SPACE_TAPE_UNIT.

Values:

Specify an integer value.

Default:

3145680

MAX_PRIQTY_IN_KB

Specifies the maximum amount of primary space that can be allocated to DASD. This parameter applies only to new copy and unload data sets.

Values:

Specify a valid PRIQTY value. You can specify the following values:

- A number that indicates the number of space units specified.
- Blank, which causes the kilobyte value shown to be converted to a value that is measured in terms of the space specified.
- 99999999, which indicates the maximum space allowed by MVS for the space unit that is specified.

Default:

3145680

MTC_CENTRAL_LOCATION

Specifies the location to use to look for changes when analyzing a multi-target change. If you do not specify a value, CM batch looks for the specified change at the current location.

To use MTC_CENTRAL_LOCATION, a DRDA connection is required between the central and remote subsystems.

Values:

Specify a valid Db2 location name.

Default:

blank

NEW_BASE_VERSION_NAME

Specifies the default name for a new base version. If this parameter is not blank, this parameter determines the default name for a new base version. Otherwise, the NEW_BASE_VERSION_NAME parameter determines the name for a new base version.

If a value is specified for a more specific base version type, for example: **BASE_VERSION_NAME_BEFORE_RUN**, that value is used for that base version type instead of the value specified for **NEW_BASE_VERSION_NAME**.

Base version parameter hierarchy:

- BASE_VERSION_OWNER
  - NEW_BASE_VERSION_OWNER
    - BASE_VERSION_OWNER_BEFORE_RUN
    - BASE_VERSION_OWNER_AFTER_RUN
- BASE_VERSION_NAME
  - NEW_BASE_VERSION_NAME
    - BASE_VERSION_NAME_BEFORE_RUN
    - BASE_VERSION_NAME_AFTER_RUN

Values:

1 to 128 characters

Default:

BASE_VERSION_NAME

NEW_BASE_VERSION_OWNER

Specifies the default owner for a new base version. If this parameter is not blank, this parameter determines the default name for a new base version. Otherwise, the NEW_BASE_VERSION_NAME parameter determines the owner for a new base version.

If a value is specified for a more specific base version type, for example, **BASE_VERSION_OWNER_BEFORE_RUN**, that value is used for that base version type instead of the value that is specified for **NEW_BASE_VERSION_OWNER**.

Base version parameter hierarchy:

- BASE_VERSION_OWNER
  - NEW_BASE_VERSION_OWNER
    - BASE_VERSION_OWNER_BEFORE_RUN
    - BASE_VERSION_OWNER_AFTER_RUN
- BASE_VERSION_NAME
  - NEW_BASE_VERSION_NAME
    - BASE_VERSION_NAME_BEFORE_RUN
    - BASE_VERSION_NAME_AFTER_RUN

Values:

1 to 128 characters

Default:

BASE_VERSION_OWNER

NEW_CHANGE_NAME

Specifies the name for a new change. If this parameter is not blank, this parameter determines the name for a new change. Otherwise, the **CHANGE_NAME** parameter determines the name for a new change.

Values:

Specify a 1- to 128-character change name or leave this parameter blank.

Default:
blank, which results in the value of the **CHANGE_NAME** parameter being used as the name for the new change.

NEW_CHANGE_OWNER
Specifies the owner for a new change. If this parameter is not blank, this parameter determines the owner for a new change. Otherwise, the **CHANGE_OWNER** parameter determines the owner for a new change.

Values:
Specify a 1- to 128-character change owner or leave this parameter blank.

Default:
blank, which results in the value of the **CHANGE_OWNER** parameter being used as the name for the new change owner.

NEW_IGNORE_NAME
Specifies the name for a new ignore. If this parameter is not blank, it determines the name for a new ignore. Otherwise, the **IGNORE_NAME** parameter determines the name for a new ignore.

Values:
Specify a 1- to 128-character ignore name or leave this parameter blank.

Default:
blank, which results in the value of the **IGNORE_NAME** parameter being used as the name for the new ignore.

NEW_IGNORE_OWNER
Specifies the owner for a new ignore. If this parameter is not blank, it determines the owner for a new ignore. Otherwise, the **IGNORE_OWNER** parameter determines the owner for a new ignore.

Values:
Specify a 1- to 128-character ignore owner or leave this parameter blank.

Default:
blank, which results in the value of the **IGNORE_OWNER** parameter being used as the name for the new ignore owner.

NEW_MASK_NAME
Specifies the name for a new mask. If this parameter is not blank, it determines the name for a new mask. Otherwise, the **MASK_NAME** parameter determines the name for a new mask.

Values:
Specify a 1- to 128-character mask name or leave this parameter blank.

Default:
blank, which results in the value of the **MASK_NAME** parameter being used as the name for the new mask name.

NEW_COMPARE_MASK_OWNER
Specifies the owner for a new mask. If this parameter is not blank, it determines the owner for a new mask. Otherwise, the **COMPARE_MASK_OWNER** parameter determines the owner for a new mask.

Values:
Specify a 1- to 128-character mask owner or leave this parameter blank.

Default:
blank, which results in the value of the **COMPARE_MASK_OWNER** parameter being used as the name for the new mask owner.

OVR_CONFIGDB_ERROR
Specifies whether Db2 Admin Tool should continue processing when change information is unable to be stored in the InfoSphere® Optim Configuration Manager repository database or the backup tables on the local system. This option applies only if integration with InfoSphere Optim Configuration Manager is enabled and the action on error setting is set to allow the override parameter.

Values:

YES

If integration with InfoSphere Optim Configuration Manager (OCM) is enabled and the action on error setting is set to allow the override parameter, Db2 Admin Tool continues processing the change even if the OCM repository database and the backup tables on the local system are not available.

NO

If integration with InfoSphere Optim Configuration Manager (OCM) is enabled, Db2 Admin Tool stops processing the change if the OCM repository database and the backup tables on the local system are not available.

Default:

NO

PDS_FOR_RECOVER_JCL

Specifies the name of a PDS to store the generated recover jobs.

Values:

A valid PDS data set name

Specify a 1- to 46-character data set name. If this name is not enclosed in single quotation marks, the fully qualified data set name is **PREFIX_FOR_DATA_SETS**.**PDS_FOR_RECOVER_JCL**.

Default:

&**SSID**..RECOVER.JCL

PDS_FOR_RECOVER_WSL

Specifies the name of a PDS to store the work statement lists (WSLs) that the analyze job generates for the recover change.

Values:

A valid PDS data set name

Specify a 1- to 46-character data set name. If this name is not enclosed in single quotation marks, the fully qualified data set name is **PREFIX_FOR_DATA_SETS**.**PDS_FOR_RECOVER_WSL**.

Default:

&**SSID**..RECOVER.WSL

PDS_FOR_RUN_JCL

Specifies the name of a PDS to store the generated run jobs, or, if ACTION_GENERATE_JCL_FROM_WSL = 'Y', a PDS to store the JCL that is generated from a WSL.

Values:

A valid PDS data set name

Specify a 1- to 46-character data set name. If this name is not enclosed in single quotation marks, the fully qualified data set name is **PREFIX_FOR_DATA_SETS**.**PDS_FOR_RUN_JCL**.

Default:

&**SSID**..RUN.JCL

PDS_FOR_RUN_JOB_INPUT

Specifies the name of a PDS in which the run job or recover job input data is stored. This parameter is used only when **USE_PERMANENT_DATA_SET_FOR_RUN_JOB_INPUT** is set to Y. You must ensure the same run job input PDS is not used for different changes. Using the same run job input PDS for different changes can cause problems when a change is run.

Values:

A valid PDS data set name

Specify a 1- to 46-character data set name. If this name is not enclosed in single quotation marks, the fully qualified data set name is **PREFIX_FOR_DATA_SETS**.**PDS_FOR_RUN_JOB_INPUT**.

Default:
&**SSID**..&CHGTAG..IN

PDS_FOR_WSL
Specifies the name of the PDS to store the work statement list (WSL) that the analyze job generates for the change.

Values:

A valid PDS data set name
Specify a 1- to 46-character data set name. If this name is not enclosed in single quotation marks,, the fully qualified data set name is **PREFIX_FOR_DATA_SETS**.**PDS_FOR_WSL**.

Default:
&**SSID**..RUN.WSL

PDS_FOR_WSL_CONV
Specifies the name of the PDS that has or will have the readable work statement list (WSL). If ACTION_CONVERT_TO_READ_WSL= 'Y' is specified, the readable WSL will be generated. If ACTION_CONVERT_TO_ISPF_WSL = 'Y' is specified, the readable WSL must already exist.

Values:
Valid PDS data set names. 1 - 46 characters. If the name is not enclosed in single quotation marks, the fully qualified data set name is PREFIX_FOR_DATA_SETS.PDS_FOR_WSL_CONV.

Default:
blank

PERCENT_INCREASE_FOR_CONVERTED_DATA_SETS
Specifies the percentage increase in size of the converted unload data set over the unload data set. The ALT/Object Compare process converts data from the UNLOAD step. The newly converted data might require more space than the unload data set. This parameter allows you to increase the size of the converted data set by a percentage greater than the unload data set, therefore helping to avoid out-of-space conditions.

Values:
A number in the range 0-100.

Default:
0

PLAN
Specifies the Db2 plan name to connect with.

Values:
Specify a 1- to 8-character Db2 plan name.

Default:
ADB

PREFIX_FOR_DATA_SETS
Specifies the data set prefix that is used when data sets are allocated, such as: WSL PDS, JCL PDS, UNLOAD, LOAD, and so on.

Values:
Specify a 1- to 17-character data set prefix.

Default:
&USERID.

PRESERVE_ALL_DATA
Specifies whether the change is to be implemented by a series of alter operations or by dropping and recreating the table. This option applies only to those changes that alter the NULLS value of a column that is at the end of the column list and is type VARCHAR.

Values:

YES
The table is dropped and recreated (by using UNLOAD, DROP, CREATE, and LOAD operations) so that the column data is preserved.

NO
The column is altered by using ALTER TABLE statements, which results in data loss for the affected columns. If you want to save a copy of the data, specify UNLOAD_ALTERED_TABLES = YES.

Default:
YES

PROCESSING_ORDER
Specifies the order that the objects will be processed in.

Values:

T
Object type processing will be performed. This process compares one object type at a time.

H
Database hierarchy processing will be performed. This process compares all object types within a database hierarchy at the same time. This is the default value.

Note: If you choose not to disable REORG optimization, the compare process will set your PROCESSING_ORDER value to H and continue processing using REORG optimization.

Default:
H

RECOVER_CHANGE_COMMENT
Specifies the comment for a recover change.

Values:
Specify a 1- to 128-character comment or leave this field blank.

Default:
blank

RECOVER_CHANGE_NAME
Specifies the name for a new recover change. If this parameter is not blank, this parameter determines the name for a new recover change. Otherwise, the name for a new recover change is the original change name with _RCVR appended.

Values:
Specify a 1- to 128-character change name.

Default:
The name of original change with _RCVR appended

RECOVER_CHANGE_OWNER
Specifies the owner for a new recover change. If this parameter is not blank, this parameter determines the owner for a new recover change. Otherwise, the owner for a new recover change is the same owner as its original change.

Values:
Specify a 1- to 128-character change owner.

Default:
The owner of original change

RECOVER_OPTION
Specifies the type of recover to be performed by the recover change process if DATA_TO_RECOVER = ‘B’ or if RECOVER_WITHDDL= ‘Y’. If neither of these conditions are true, leave RECOVER_OPTION blank.

Values:

O

Recover using the original data. The *original data* is the data that is unloaded when the original change is run. If you use the original data during a recovery operation, consider whether related tables that were not affected by the recover must also be restored to the same point to avoid inconsistencies. This option applies only to tables that were unloaded in the original change.

E

Recover using the existing data. The *existing data* is the data that exists in the table just before the original change is recovered. If a table is dropped without being re-created in the original change, no data is loaded after the table is created in the recover change.

D

Recover with DDL only and ignore data.

Default:

blank

Recover using the option that was specified when the recover change was set up. If DATA_TO_RECOVER= 'B' or RECOVER_WITHDDL = ‘Y’, blank is not valid; you must specify a value for RECOVER_OPTION.

RECOVER_PENDING_CHANGE_ACTION

Specifies the action that occurs if the change being recovered contains changes to objects that have changes pending from Db2 Admin Tool Change Management.

Values:

S

This option recovers the specified change and set to DEFINED status for any pending change that modifies the same or related objects. The recover change supersedes any pending changes that modify the same or related objects.

C

This option prevents the change from being recovered when pending changes will modify the same or related objects. If there are pending changes, the changes are not recovered. To recover this change and to set the status of any pending changes to DEFINED, set the value of this parameter to S (supersede).

Default:

C

RECOVER_WITHDDL

Specifies whether recover change can recover using DDL only and no data.

Values:

Y

Recover change can recover using DDL only and ignore data.

N

Recover change can not recover using DDL only.

Default:

N

RECREATE_ACCELERATED_TABLES

Specifies whether to automatically detect and recreate accelerated tables, in DROP/CREATE or ALTER situations, involving changes to its definition or partitions.

Values:

YES

Recreate accelerated tables.

NO

Do not recreate accelerated tables.

Default:
YES

If RECREATE_ACCELERATED_TABLES is set to NO, the following related options must also be set to NO:

- ENABLE_ACCELERATED_TABLES
- REPLICATE_ACCELERATED_TABLES
- RELOAD_ACCELERATED_TABLES

RELOAD_ACCELERATED_TABLES
Specifies whether to automatically detect and reload accelerated tables in DROP/CREATE or ALTER situations of accelerated tables, involving changes to its definition, data or partitions.

Y
Reload accelerated tables.

N
Do not reload accelerated tables.

Default:
Y

If RELOAD_ACCELERATED_TABLES is set to YES, RECREATE_ACCELERATED_TABLES must also be set to YES.

If RELOAD_ACCELERATED_TABLES is set to NO, the following related options must also be set to NO:

- ENABLE_ACCELERATED_TABLES
- REPLICATE_ACCELERATED_TABLES

REMOVE_ACCELERATED_TABLES
Specifies whether to automatically detect and remove dropped Db2 tables involved in change management from the accelerator.

Values:

YES
Remove accelerated tables.

NO
Do not remove accelerated tables.

Default:
YES

REPLICATE_ACCELERATED_TABLES
Specifies whether to enable replication for accelerated tables that are automatically re-created and reloaded in DROP/CREATE or ALTER situtations.

Values:

Y
Replicates accelerated tables.

N
Does not replicate accelerated tables.

Default:
Y

If REPLICATE_ACCELERATED_TABLES is set to YES, RELOAD_ACCELERATED_TABLES must also be set to YES.

REPORT_EXPECTED_CONVERSION_PROBLEMS
Specifies whether a report is generated of the data conversion problems for tables that are expected to occur when the change is run.

Values:

Y

The report includes the expected conversion problems for tables when the change is run.

N

The report does not include a list of expected conversion problems.

Default:

N

REPORT_OBJECT_COUNT

Specifies whether a statistics report is generated of compared and changed objects for each object type.

Values:

Y

The report includes statistics of compared and changed objects for each object type.

N

The report does not include the object count statistics.

Default:

N

REPORT_OBJECT_TOTALS

Specifies whether a summary report of the overall totals for each object type is generated. The number of altered, created, dropped, or recreated objects for each object type is included.

The Totals Report contains the overall totals of objects affected and includes values from the Analyze step from each of the target systems.

Note: Specifying REPORT_OBJECT_TOTALS = 'Y' forces REPORT_OBJECT_COUNTS to be set to 'Y'.

Values:

Y

The report includes the object totals report.

N

The report does not include the object totals report.

Default:

N

REPORT_ONLY_CHANGED_OBJECTS

Specifies whether to report objects which are identical in the source and the target.

Values:

Y

The report does not include objects that are identical in the source and target.

N

The report includes objects that are identical in the source and the target.

Default:

N

REPORT_STMT_TOTALS

Specifies whether a totals report of totals for each statement type is generated during the Run phase. The number of ALTER, CREATE, or DROP statements issued for each object type is included.

Values:

Y

The report includes the statement totals report.

N

The report does not include the statement totals report.

Default:
N

REPORT_SUMMARY
Specifies whether to include a brief summary of changes for each object in the report.

Values:

Y
The report includes a brief summary of changes for each object.

N
The report does not include a brief summary.

Default:
N

REPORT_SYSTEM_GENERATED_IGNORE_FIELDS
Specifies whether to include in the report the system generated names of the fields that are ignored.

Values:

Y
The report includes system generated names of the fields that are ignored by the compare or analyze process.

N
The report does not include system generated names of the fields that are ignored.

Default:
N

REPORT_TRANSLATION_MASKS
Specifies whether to report the translation masks that are used.

Values:
Y, N

Y
The report includes the masks used by the compare or analyze process.

N
The report does not include the masks that are used.

Default:
N

REPORT_USER_SPECIFIED_IGNORE_FIELDS
Specifies whether the report includes the user-defined names of the fields that are ignored.

Values:

Y
The report includes user-defined names of the fields that are ignored by the compare or analyze process.

N
The report does not include user-defined names of the fields that are ignored.

Default:
N

RETAIN_GENERATED_ALWAYS_FOR_ROW_CHANGE_TS
Specifies whether to retain GENERATED ALWAYS for the designated column types (ROWID or ROW CHANGE TIMESTAMP).

Values:

Y
Retain the GENERATED ALWAYS attribute for row change time stamp columns.

N

Do not retain the GENERATED ALWAYS attribute for row change time stamp columns.

Default:

N

RETAIN_GENERATED_ALWAYS_FOR_ROWID

Specifies whether to retain GENERATED ALWAYS for the designated column types (ROWID or ROW CHANGE TIMESTAMP).

Values:

Y

Retain the GENERATED ALWAYS attribute for rowid columns.

N

Do not retain the GENERATED ALWAYS attribute for rowid columns.

Default:

N

RUN_CHECK_DATA

Specifies whether to generate a CHECK DATA utility job for the table spaces that are affected by the (RE)LOAD utility jobs that the analyze process generates in the WSL.

Values:

Y

Generate a CHECK DATA utility job for each table space that is affected by a LOAD utility.

N

Do not generate a CHECK DATA utility job.

Default:

N

RUN_REBIND

Specifies whether to generate a job to rebind plans and packages that are affected by changes that are generated by Object Comparison Tool.

Values:

M

Generate REBIND statements for packages only if packages are invalidated by the changes.

A or Y

Generate REBIND statements for packages and plans that are affected by the change.

N

Do not generate REBIND statements.

Default:

N

RUN_REORG_REBUILD

Specifies whether to generate REORG table space and REBUILD index utility jobs after applying the changes from the analyze process, the purpose of which is to make the target operational.

Values:

M

Mandatory. Generate REORG utility statements to remove REORG-pending conditions.

A

All relevant. Generate all needed REORG utility statements to fully implement the effects of the changes, for example, space parameter changes.

N

None. No REORG utility statements are generated. This option is invalid if you specified NO for ALLOW_ROTATE_PARTS.

Default:
N

RUN_RUNSTATS
Specifies whether to generate a RUNSTATS utility job for the table spaces that are affected by the RE(LOAD) utility jobs and for the table spaces, tables, and indexes that are affected by SQL ALTER statements that the analyze process generates in the WSL.

Values:

R
Generate RUNSTATS utility statements for all tables that are affected by the (RE)LOAD utility.

A
Generate RUNSTATS utility statements for all altered table space, table, and index objects.

B
Generate RUNSTATS utility statements for objects that are affected by the RE(LOAD) utility and SQL ALTER statements.

M
Generate RUNSTATS utility statements for the following conditions:

- If a table space is dropped and recreated, the tables and indexes are included in RUNSTATS.
- If a table is dropped and recreated, only the indexes, and not the table, are included in RUNSTATS.
- If an index is created, recreated, or has columns added, the index is included in RUNSTATS.
- If the index is created with DEFER YES and REBUILD is generated, RUNSTATS is performed after the REBUILD.

N
No RUNSTATS utility statements are generated.

Default:
N

RUN_SQLID
Specifies whether SET CURRENT SQLID statements are generated and, if so, what SQLID value to use.

Values:

An SQLID
The specified Run SQLID is the owner of databases and table spaces. If the specified Run SQLID is different from the current owner, the databases, table spaces, and all dependent objects are dropped and re-created to accomplish the change of owner.

<NONE>
No SET CURRENT SQLID statements are generated.

blank
SET CURRENT SQLID statements are generated when necessary.

Default:
blank

SAVE_COMPARE_RESULTS
Specifies whether compare results are saved during the compare run. You can manage the saved compare result using the "MR - Manage saved compare results" dialogs, which you access from the Object Comparison Tool main menu.

Values:
YES, NO

Default:
NO

SAVE_SOURCE_BASE_VERSION

Specifies whether to save the source base version that is generated for the change during the analyze process. The source base version represents the Db2 object definitions after the change is implemented.

Values:

Y

The source base version generated during analyze is saved as a new base version.

N

The source base version generated during analyze is not saved.

Default:

N

SAVE_TARGET_BASE_VERSION

Specifies whether to save the target base version that is generated for the change during the analyze process. The target base version represents the Db2 object definitions as they existed in the Db2 catalog at analyze time, with Db2 Admin Tool change management pending changes applied, but without the changes for the specified change applied.

Values:

Y

The target base version that was generated during analyze is saved as a new base version.

N

The source base version during analyze is not saved.

Default:

N

SCOPE_WARNING

Specifies whether to issue a warning message if 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.

Values:

Y

Message ADB7353W is issued for this situation. This message is issued regardless of the value of the value of the **Suppress DROP of objects** field on the **Generate Compare Jobs (GOC5)** panel.

N

No message is issued.

Default:

N

SEQUENCE_RESTART_VALUE

Specifies what the value for the RESTART attribute is when a Db2 sequence object is re-created. Use this parameter only for recovery paths.

Values:

ORIGINAL, COMPUTED

Default:

ORIGINAL

SOURCE_DSN

Specifies the name of the data set that contains the compare source. Specifying this parameter overrides a pre-allocated compare source input file (SRCIN DD).

Values:

A data set name

Specify a 1- to 46-character data set name. If **SOURCE_TYPE** = 'DDL', specify the name of the data set that contains the DDL for the compare source.

If **SOURCE_TYPE** = 'USER', specify the name of the data set that contains the list of Db2 Admin Tool quick scopes for the compare source.

blank

If **SOURCE_TYPE** = 'DDL', the SRCIN file must contain the DDL for the compare source.

If **SOURCE_TYPE** = 'USER', either the **SOURCE_VERSION_SCOPE_OWNER** and **SOURCE_VERSION_SCOPE_NAME** parameters must be specified, or the SRCIN file must contain the list of Db2 Admin Tool quick scopes for the compare source.

Default:

blank

SOURCE_EXCLUDE_NAME

Specifies the name of an Exclude Specification that is stored in the Change Management database. The **SOURCE_EXCLUDE_OWNER** and **SOURCE_EXCLUDE_NAME** parameters identify an existing Exclude Specification to be used for the compare source.

Values:

A valid exclude specification name, blank

A valid exclude specification name.

Specify a 1- to 128-character exclude specification name. The specified Exclude Specification is used for the source during the compare process.

blank

Exclude objects are not used for the compare source.

Default:

blank

SOURCE_EXCLUDE_OWNER

Specifies the owner of an Exclude Specification that is stored in the Change Management database. The **SOURCE_EXCLUDE_OWNER** and **SOURCE_EXCLUDE_NAME** parameters identify an existing Exclude Specification to be used for the compare source.

Values:

Specify a valid 1- to 128-character exclude specification owner.

Default:

&CURSQLID.

SOURCE_GEN_FOREIGN_KEYS

Specifies whether to generate DDL for ALTER TABLE ADD FOREIGN KEY for the source objects. SOURCE_GEN_FOREIGN_KEYS is valid only for SOURCE_TYPE = 'USER'.

Values:

Y

Generate the DDL.

D

Generate the DDL and extract foreign keys for tables that are dependent on the tables being extracted.

N

Do not generate DDL.

Default:

blank

SOURCE_LOCATION

Specifies the Db2 location for the compare source when the Db2 objects are located in a Db2 subsystem.

Values:
Specify a valid 1- to 128-character location name that is defined in SYSIBM.LOCATIONS or leave this parameter blank to specify the local Db2 subsystem.

blank
The local Db2 subsystem.

Default:
blank

SOURCE_TYPE
Specifies the type of input that identifies the Db2 objects for the source of the compare.

Values:

DDL
The source is DDL. To specify the data set that contains the DDL, use the compare source input file (SRCIN DD) or the **SOURCE_DSN** parameter. If the **SOURCE_DSN** parameter is not specified, the compare source input file (SRCIN DD) must be pre-allocated.

USER
The source is a Db2 subsystem, and the list of object names is provided by the user. To specify the list of Db2 objects, use a Db2 Admin Tool version scope, a list of Db2 Admin Tool quick scopes, or both.

To specify an existing scope, use the **SOURCE_VERSION_SCOPE_OWNER** and **SOURCE_VERSION_SCOPE_NAME** parameters. To specify a data set that contains a list of Db2 Admin Tool quick scopes, use the compare source input file (SRCIN DD) or the **SOURCE_DSN** parameter.

Default:
DDL

Related information:

"Version scopes" on page 865
"Quick scopes" on page 869

SOURCE_VERSION_COMMENT
Specifies a comment or description of the source version.

Values:
Specify a 1- to 128-character comment, or leave this field blank.

Default:
blank

SOURCE_VERSION_NAME
Specifies the name for the base version that will store the generated source base version work file. If the **SOURCE_VERSION_OWNER** and **SOURCE_VERSION_NAME** parameters identify an existing base version, the **EXISTING_BASE_VERSION_ACTION** parameter controls whether the existing base version is replaced or a product-generated version name is used.

Values:
Specify a valid 1- to 128-character version name.

Default:
AUTO:OC.&CURTS..SRCVF

SOURCE_VERSION_OWNER
Specifies the owner for the base version that will store the generated source base version work file. If the **SOURCE_VERSION_OWNER** and **SOURCE_VERSION_NAME** parameters identify an existing base version, the **EXISTING_BASE_VERSION_ACTION** parameter controls whether the existing base version is replaced or a product-generated version name is used.

Values:
Specify a valid 1- to 128-character version owner.

Default:
&CURSQLID.

SOURCE_VERSION_SCOPE_NAME
Specifies the name of the version scope for the compare source. It is only used if the source type is USER.

Values:
A valid version scope name; 1 to 128 characters, blank

A valid version scope name.
Specify a valid 1- to 128-character version scope name. If SOURCE_TYPE = 'USER', the version scope that is specified by **SOURCE_VERSION_SCOPE_OWNER** and **SOURCE_VERSION_SCOPE_NAME** is used for the Db2 object list for the compare source.

blank
If SOURCE_TYPE = 'USER', a list of Db2 Admin Tool quick scopes must be specified in a pre-allocated SRCIN DD file or in the data set that is specified by the **SOURCE_DSN** parameter.

Default:
blank

SOURCE_VERSION_SCOPE_OWNER
Specifies the owner of the version scope for the compare source. This parameter is used only if the source type is USER.

Values:
Specify a valid 1- to 128-character version scope owner.

Default:
&CURSQLID.

SOURCE_VERSION_TYPE
Specifies the final disposition of the generated source base version work file. If the SRCVF file is pre-allocated, this parameter has no effect for types FILE and TEMP.

Values:

FILE
If the SRCVF file is not pre-allocated, it is allocated by using the attributes from **ADMIN_DATASET_TYPE** = 'SRCVF'.

DB2
If the SRCVF file is not pre-allocated, it is allocated by using the attributes from **ADMIN_DATASET_TYPE** = 'SRCVF' but as a temporary file. The file contents are stored in the Db2 Admin Tool change management repository using the owner and name values from the **SOURCE_VERSION_OWNER** and **SOURCE_VERSION_NAME** parameters.

TEMP
If the SRCVF file is not pre-allocated, it is allocated by using the attributes from **ADMIN_DATASET_TYPE** = 'SRCVF' but as a temporary file.

Default:
FILE

SPACE_TAPE_UNIT
Specifies the name of a valid tape unit. This parameter applies only to new copy and unload data sets.

Values:
Specify a valid space unit for tape.

Default:
TAPE

SPACE_UNIT
Specifies the units in which new data sets are to be allocated. This parameter applies only to new copy and unload data sets. Specifying BLK causes Db2 Admin Tool to allocate in blocks of 8192 bytes, which is the block size used by the Db2 Unload utility.

Values:
Specify a valid space unit.

Default:
TRK

SPACE_UNIT_NAME
Specifies the default unit name.

Values:
Specify a valid space unit name.

Default:
SYSALLDA

SSID
Specifies the Db2 subsystem to connect to.

Values:
Specify a valid 1- to 4-character Db2 subsystem ID.

Default:
This parameter does not have a default value.

STOP_ON_CONVERSION_ERROR
Specifies whether to stop WSL processing when data conversion errors occur.

Values:

Y
Stop WSL processing with RC=28 when conversion errors occur.

N
Do not stop WSL processing when conversion errors occur.

Default:
N

SUPPRESS_ADDING_COLUMNS
Specifies whether compare should suppress adding target columns.

Values:
YES, NO

Default:
NO

SUPPRESS_DROP_OF_COLUMNS
Specifies whether compare should suppress dropping target columns.

Values:
YES, NO

Default:
NO

SUPPRESS_DROP_OF_OBJECTS
Specifies whether the compare process suppresses dropping target objects that are in the target but that are not in the source.

Values:
YES, NO

Default:
NO

Regardless of the value that you set for this option, IBM Db2 Object Comparison Tool for z/OS replaces all relationships between a parent and a child if a foreign key is specified in the source. To delete a foreign key, both the parent and the child must be present in the source (without a foreign key). If DROP statements are part of the source DDL, objects are dropped regardless of the value that is specified for this parameter.

Regardless of the value that you set for this option, IBM Db2 Object Comparison Tool for z/OS drops all explicit LOB objects from the target if they are not specified on the source. However, if the base table that is associated with the LOB objects is kept because 'Suppress DROP of objects' is set to 'YES', then all of the LOB objects are kept.

Note: If the TARGET_TYPE = 'AUTO' for Target is used, the **`SUPPRESS_DROP_OF_OBJECTS`** parameter is forced to a setting of YES. If NO was specified, a warning message is issued stating that the change was made.

SYMBOL_NAME

Specifies the name of a user-defined symbol variable to use to mask some of the parameter values at run time.

Values:

a valid symbol variable name

Specify a valid symbol variable name or leave this parameter blank. A valid symbol variable name begins with the ampersand (&) character and ends with the `.` character. The name can be 3-128 characters, the total of which includes the & and `.` characters. The name is converted to upper case.

Default:

blank

SYMBOL_VALUE

Specifies the value of a user-defined symbol variable to be used to mask some of the parameter values at run time.

Values:

Specify a 1- to 128-character value or leave this field blank.

Default:

blank

TAKE_AN_IMAGE_COPY

Specifies whether to generate a COPY utility job for the table spaces that are affected by the RE(LOAD) utility jobs and for the table spaces, tables, and indexes that are affected by SQL ALTER statements that the analyze process generates in the WSL.

Values:

R

Generate COPY utility statements for all tables that are affected by the (RE)LOAD utility.

A

Generate COPY utility statements for all altered table space, table, and index objects.

B

Generate COPY utility statements for objects that are affected by the RE(LOAD) utility and SQL ALTER statements.

N

No COPY utility statements are generated.

Default:

N

TARGET_ASSOCIATIONID

Specifies the association ID provided by multi-target central system used to identify the target change.

Values

The value originates from the multi-target change file, which cannot be modified by the user.

Default:

blank

TARGET_CHANGE_COMMENT

Specifies the comment for a new change on the target system.

Values:
1 to 128 characters

Default:
blank

TARGET_DSN
Specifies the name of the data set that contains the compare target. This parameter is used when the TARGET_TYPE is USER. Specifying this parameter overrides a pre-allocated compare target input file (TGTIN DD).

Values:

A data set name.
Specify a 1- to 46-character data set name. If TARGET_TYPE = 'USER', specify the name of the data set that contains the list of Db2 Admin Tool quick scopes for the compare target; One to 46 characters

blank
If TARGET_TYPE = 'USER', either the **TARGET_VERSION_SCOPE_OWNER** and **TARGET_VERSION_SCOPE_NAME** parameters must be specified, or the TGTIN file must contain the list of Db2 Admin Tool quick scopes for the compare target.

Default:
blank

TARGET_EXCLUDE_NAME
Specifies the name of an Exclude Specification that is stored in the Change Management database. The **TARGET_EXCLUDE_OWNER** and **TARGET_EXCLUDE_NAME** parameters identify an existing Exclude Specification to be used for the compare target.

Values:

A valid exclude specification name
Specify a valid 1- to 128-character exclude specification name. The specified Exclude Specification is used for the target during the compare process. One to 128 characters

blank
Exclude objects is not used for the compare target.

TARGET_EXCLUDE_OWNER
Specifies the owner of an Exclude Specification that is stored in the Change Management database. The **TARGET_EXCLUDE_OWNER** and **TARGET_EXCLUDE_NAME** parameters identify an existing Exclude Specification to be used for the compare target.

Values:
Specify a valid 1- to 128-character exclude specification owner.

Default:
&CURSQLID.

TARGET_GEN_FOREIGN_KEYS
Specifies whether to generate DDL for ALTER TABLE ADD FOREIGN KEY for the target objects. TARGET_GEN_FOREIGN_KEYS is valid only for TARGET_TYPE = 'AUTO'.

Values:

Y
Generate the DDL.

D
Generate the DDL and extract foreign keys for tables that are dependent on the tables being extracted.

N
Do not generate DDL.

Default:
blank

TARGET_IGNORE_NAME

Specifies the name of an existing Ignore Fields entry as defined in the Change Management database on the target system. The Ignore Fields entry on the target system, that is identified by the **TARGET_IGNORE_OWNER** and TARGET_IGNORE_NAME parameters, is used to ignore the Db2 columns when the change on the target system is analyzed.

Values:

A valid Ignore Fields name

Specify a valid 1- to 128-character ignore fields name.

blank

Ignore Fields name is not included in the statement.

Default:

blank

TARGET_IGNORE_OWNER

Specifies the owner of an existing Ignore fields entry defined in the Change Management database on the target system. The Ignore Fields entry on the target system, that is identified by the **TARGET_IGNORE_OWNER** and TARGET_IGNORE_NAME parameters, is used to ignore the Db2 columns when the change on the target system is analyzed.

Values:

A valid Ignore Fields owner

Specify a valid 1- to 128-character target ignore owner.

blank

Ignore Fields owner is not included in the statement.

Default:

blank

TARGET_LOCATION

Specifies the Db2 location for the compare target when the Db2 objects are located in a Db2 subsystem. If the compare result is imported as a new change (**ACTION_IMPORT_CHANGE** = 'Y') the target location must be the local Db2 subsystem.

Values:

Specify a location that is defined in SYSIBM.LOCATIONS or leave this field blank to specify the local Db2 subsystem.

Default:

blank

TARGET_MASK_NAME

Specifies the name of an existing mask defined in the Change Management database on the target system. The mask on the target system, that is identified by the **TARGET_COMPARE_MASK_OWNER** and **TARGET_MASK_NAME** parameters, is used to mask the change statements when the change on the target system is registered.

Values:

A valid mask name

Specify a valid 1- to 128-character target mask name.

blank

The mask name is not included in the statement.

Default:

blank

TARGET_COMPARE_MASK_OWNER

Specifies the owner of an existing mask defined in the Change Management database on the target system. The mask on the target system, that is identified by the **TARGET_COMPARE_MASK_OWNER** and **TARGET_MASK_NAME** parameters, is used to mask the change statements when the change on the target system is registered.

Values:

A valid mask owner
Specify a valid 1- to 128-character target mask owner.

blank
The mask owner is not included in the statement.

Default:
blank

TARGET_TYPE
Specifies the type of input that identifies the Db2 objects for the target of the compare.

Values:

AUTO
The target is a Db2 subsystem. The Db2 objects for the compare target are automatically selected by Db2 Admin Tool based on the content of the compare source.

If TARGET_TYPE=AUTO, you cannot specify TARGET_DSN, TARGET_VERSION_SCOPE_NAME, or a compare target input file (TGTIN DD).

USER
The target is a Db2 subsystem, and the list of object names is provided by the user. To specify the list of Db2 objects, use a Db2 Admin Tool version scope, a list of Db2 Admin Tool quick scopes, or both.

To specify an existing scope, use the **TARGET_VERSION_SCOPE_OWNER** and **TARGET_VERSION_SCOPE_NAME** parameters. To specify a data set that contains a list of Db2 Admin Tool quick scopes, use the compare target input file (TGTIN DD) or the **TARGET_DSN** parameter.

DDL
The target is DDL. To specify the data set that contains the DDL, use the compare target input file (TGTIN DD) or the **TARGET_DSN** parameter. If the **TARGET_DSN** parameter is not specified, the compare target input file (TGTIN DD) must be pre-allocated.

⚠ **Attention:** If the target DDL that is specified does not match the DDL in the Db2 database, problems might occur when running the resultant change. Therefore, consider using this option only when comparing DDL to DDL (TARGET_TYPE='DDL' and SOURCE_TYPE='DDL').

Default:
AUTO

Related information:

"Version scopes" on page 865
"Quick scopes" on page 869

TARGET_VERSION_COMMENT
Specifies a comment or description of the target version.

Values:
Specify a 1- to 128-character comment or leave this field blank.

Default:
blank

TARGET_VERSION_NAME
Specifies the name for the base version that will store the generated target base version work file. If the **TARGET_VERSION_OWNER** and **TARGET_VERSION_NAME** parameters identify an existing base version, the **EXISTING_BASE_VERSION_ACTION** parameter controls whether the existing base version is replaced or a product-generated version name is used.

Values:
Specify a valid 1- to 128-character version name.

Default:
AUTO:OC.&CURTS..TGTVF

TARGET_VERSION_OWNER
Specifies the owner for the base version that will store the generated target base version work file. If the **TARGET_VERSION_OWNER** and **TARGET_VERSION_NAME** parameters identify an existing base version, the **EXISTING_BASE_VERSION_ACTION** parameter controls whether the existing base version is replaced or a product-generated version name is used.

Values:
Specify a valid 1- to 128-character version owner.

Default:
&CURSQLID.

TARGET_VERSION_SCOPE_NAME
Specifies the name of the version scope for the compare target. It is only used if the target type is USER.

Values:

A valid version scope name.
Specify a valid 1- to 128-character version scope name. If TARGET_TYPE = 'USER', the version scope that is specified by **TARGET_VERSION_SCOPE_OWNER** and **TARGET_VERSION_SCOPE_NAME** is used for the Db2 object list for the compare target.

blank
If **TARGET_TYPE = 'USER'**, a list of Db2 Admin Tool quick scopes must be specified in a pre-allocated TGTIN DD file or in the data set specified by the **TARGET_DSN** parameter.

Default:
blank

TARGET_VERSION_SCOPE_OWNER
Specifies the owner of the version scope for the compare target. This parameter is used only if the TARGET_TYPE parameter is set to USER.

Values:
Specify a valid 1- to 128-character version scope owner.

Default:
&CURSQLID.

TARGET_VERSION_TYPE
Specifies the final disposition of the generated target base version work file. If the TGTVF file is pre-allocated this parameter has no effect for types FILE and TEMP.

Values:

FILE
If the TGTVF file is not pre-allocated, it is allocated by using the attributes from **ADMIN_DATASET_TYPE** = 'TGTVF'.

DB2
If the TGTVF file is not pre-allocated, it is allocated by using the attributes from **ADMIN_DATASET_TYPE** = 'TGTVF' but as a temporary file. The file contents are stored in the Db2 Admin Tool change management repository. The owner and name values are obtained from the **TARGET_VERSION_OWNER** and **TARGET_VERSION_NAME** parameters.

TEMP
If the TGTVF file is not pre-allocated, it is allocated by using the attributes from **ADMIN_DATASET_TYPE** = 'TGTVF', but as a temporary file.

Default:
FILE

TGT_DB2FL

Specifies the target function level for generated DDL statements. CM batch generates DDL statements based on the syntax requirements for the specified target function level.

Values:

An integer value in the range 501 - 999 or 100. You can specify any function level equal to or less than the current Db2 function level.

Default:

The current Db2 function level

Related information:

Adopting new capabilities in Db2 12 continuous delivery (Db2 12 for z/OS)
"Support for Db2 continuous delivery" on page 69

UNLOAD_ALTERED_TABLES

Specifies whether to unload altered tables as part of the analyze process. Regardless of the UNLOAD_ALTERED_TABLES value, the table is always unloaded if the change is performed by using DROP and CREATE.

Values:

YES or Y

Unload all altered tables.

DES or D

Unload only those altered tables where the alters are destructive. *Destructive alters* are changes that might result in the loss of data, such as ALTER TABLE DROP COLUMN.

NO or N

Do not unload any altered tables.

Default:

YES

UNLOAD_METHOD

Specifies the method that is used to unload the data.

Values:

U

Use the UNLOAD utility.

P

Use the Db2 Parallel UNLOAD utility.

H

Use Db2 High Performance Unload when available. The HPU option is supported only if an HPU load library is specified.

Default:

U

USE_DEFER_YES

Specifies whether to use DEFER YES clauses on any eligible CREATE INDEX statements. Any user-specified masks will have precedence. This value is also used for subsequent runtime analysis to ensure that the same DDL and Db2 Admin Tool statements are generated.

Values:

Y

Specify DEFER YES on eligible indexes.

N

Do not specify DEFER YES.

Default:

USE_DSNUTIL_SP

Specifies whether a Db2 stored procedure is used to run utilities.

Values:

YES or Y

Utilities are run by calling DSNUTILV or DSNUTILU. If you are using Db2 12 for z/OS, DSNUTILV is used. If you are using a prior version of Db2, DSNUTILU is used.

NO or N

Utilities are invoked in the CM batch job.

Default:

YES

USE_IGNORE_FOR_IMPORT_CHANGE

Specifies whether an ignore is used for the imported change.

Values:

Y

If an ignore is also being imported, the ignore that is used for import change is the newly created ignore. Otherwise, the ignore that is used is identified by the **IGNORE_OWNER** and **IGNORE_NAME** parameters.

blank

Specifies that this parameter defaults to Y if an ignore and a change are imported.

Default:

blank

USE_MASK_FOR_EXPORT_CHANGE

Specifies whether the data for export change is masked during export.

Values:

Y

If a mask is also being exported, the mask that is used for export change is the newly created mask. Otherwise, the mask that is used is identified by the **COMPARE_MASK_OWNER** and **MASK_NAME** parameters.

blank

Specifies that this parameter defaults to Y if a mask and a change are imported.

Default:

blank

USE_MASK_FOR_IMPORT_CHANGE

Specifies whether the input for import change is masked during import.

Values:

Y

If a mask is also being imported, the mask that is used for import change is the newly created mask. Otherwise, the mask that is used is identified by the **COMPARE_MASK_OWNER** and **MASK_NAME** parameters.

blank

Specifies that this parameter defaults to Y if a mask and a change are imported.

Default:

blank

USE_PERMANENT_DATA_SET_FOR_RUN_JOB_INPUT

Specifies where to store the run job input. The run job input can be put in-stream in the run job itself, or into a PDS.

Values:

Y

Store the run job input data in a permanent data set that is referenced in the run job.

N
Store the run job input data in an in-stream data set in the run job.

Default:
N

USE_UTILITY_OPTIONS
Specifies whether to use the customized utility options. The utility options that you can specify are described in "Utility option parameters" on page 745

Values:

Y
The user-customized utility options are used.

N
The Db2 Admin Tool and Db2 default utility options are used.

Default:
N

UTIL_CLONE_TEMPLATE_COPYDDN1_NAME
Specifies the user-provided template name for the first file of COPYDDN.

Values:
Specify a 1- to 8-character Db2 template name.

Default:
CLNCOPY1

UTIL_CLONE_TEMPLATE_COPYDDN1_USE
Specifies whether to use a user-provided template for the first COPYDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_COPYDDN1_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:
a non-blank value

a non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** is set to Y, and the template exists in the ADBTEMPL file.

Default:
S

UTIL_CLONE_TEMPLATE_COPYDDN2_NAME
Specifies the user-provided template name for the second file of COPYDDN.

Values:
A Db2 template name; 1 to 8 characters

Default:
CLNCOPY2

UTIL_CLONE_TEMPLATE_COPYDDN2_USE
Specifies whether to use a user-provided template for the second COPYDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_COPYDDN2_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:
a non-blank value

a non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_CLONE_TEMPLATE_DISCARDDN_NAME

Specifies the user-provided template name for the DISCARDDN file.

Values:

A Db2 template name; 1 to 8 characters

Default:

CLNDISC

UTIL_CLONE_TEMPLATE_DISCARDDN_USE

Specifies whether to use a user-provided template for the DISCARDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_DISCARDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

UTIL_CLONE_TEMPLATE_ERRDDN_NAME

Specifies the user-provided template name for the ERRDDN file.

Values:

A Db2 template name; 1 to 8 characters

Default:

CLNERR

UTIL_CLONE_TEMPLATE_ERRDDN_USE

Specifies whether to use a user-provided template for the ERRDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_ERRDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

UTIL_CLONE_TEMPLATE_FCCOPYDDN_NAME

Specifies the user-provided template name for the FCCOPYDDN file.

Values:

A Db2 template name; 1 to 8 characters

Default:

CLNFCOPY

UTIL_CLONE_TEMPLATE_FCCOPYDDN_USE

Specifies whether to use a user-provided template for the FCCOPYDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_FCCOPYDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

a non-blank value

Default:

S

UTIL_CLONE_TEMPLATE_LOBCOL_NAME

Specifies the user provided template name for LOB columns.

Values:
A Db2 template name; 1 to 8 characters

Default:
CLNLOBC

The data set name cannot exceed 35 bytes and must be in PDS format. Do not specify a member name

UTIL_CLONE_TEMPLATE_LOBCOL_USE
Specifies whether to use a user-provided template for templates related to LOB columns. If a non-blank value is specified, the template name for LOB columns is determined from the **UTIL_CLONE_TEMPLATE_LOBCOL_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** is set to Y.

Values:
a non-blank value

Default:
S

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** is set to Y, and the template exists in the ADBTEMPL file.

UTIL_CLONE_TEMPLATE_MAPDDNNAME
Specifies the user-provided template name for MAPDDN.

Values:

A Db2 template name.
Specify a 1- to 8-character Db2 template name.

Default:
CLNMAP

UTIL_CLONE_TEMPLATE_MAPDDNUSE
Specifies whether to use a user-provided template for the MAPDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_MAPDDNNAME** parameter. This parameter is only in effect if the **GENERATE_TEMPLATES** is set to Y.

Values:

A non-blank value.
Specify a non-blank value.

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** is set to Y, and the template exists in the ADBTEMPL file.

Default:
S

UTIL_CLONE_TEMPLATE_PUNCHDDN_NAME
Specifies the user provided template name for the PUNCHDDN file of the REORG utility.

Values:
A Db2 template name; 1 to 8 characters

Default:
CPUNCH

UTIL_CLONE_TEMPLATE_PUNCHDDN_USE
Specifies whether to use a user provided template for the PUNCHDDN file of the REORG utility. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_PUNCHDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:
a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

UTIL_CLONE_TEMPLATE_RECOVERYDDN1_NAME

Specifies the user-provided template name for the first name for RECOVERYDDN.

Values:

A Db2 template name; 1 to 8 characters

Default:

CLNRCVR1

UTIL_CLONE_TEMPLATE_RECOVERYDDN1_USE

Specifies whether to use a user-provided template for the first RECOVERYDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_RECOVERYDDN1_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set toY and the template exists in the ADBTEMPL file.

Default:

S

UTIL_CLONE_TEMPLATE_RECOVERYDDN2_NAME

Specifies the user-provided template name for the second name for RECOVERYDDN.

Values:

A Db2 template name; 1 to 8 characters

Default:

CLNRCVR2

UTIL_CLONE_TEMPLATE_RECOVERYDDN2_USE

Specifies whether to use a user-provided template for the second RECOVERYDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_RECOVERYDDN2_NAME** parameter. This parameter is in effect only if **GENERATE_TEMPLATES** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

UTIL_CLONE_TEMPLATE_UNLDDN_NAME

Specifies the user provided template name for the UNLDDN file of the REORG utility.

Values:

A Db2 template name; 1 to 8 characters

Default:

CUNL

UTIL_CLONE_TEMPLATE_UNLDDN_USE

Specifies whether to use a user provided template for the UNLDDN file of the REORG utility. If a non-blank value is specified, the template name is determined from the

UTIL_CLONE_TEMPLATE_UNLDDN_NAME parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:
a non-blank value

a non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDN_NAME
Specifies the user provided template name for the PUNCHDDN file of the UNLOAD utility.

Values:
A Db2 template name; 1 to 8 characters

Default:
CUPUNCH

UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDN_USE
Specifies whether to use a user provided template for the PUNCHDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:
a non-blank value

a non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDNC_NAME
Specifies the user provided template name for the Db2 Admin Tool converted version of the PUNCHDDN file of the UNLOAD utility. Some types of changes require that the unloaded data be converted by Db2 Admin Tool before the data is loaded. This parameter controls the user provided template for the converted load control card for the unloaded data.

Values:
A Db2 template name; 1 to 8 characters

Default:
CUPUNCHC

UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDNC_USE
Specifies whether to use a user provided template for the Db2 Admin Tool converted version of the PUNCHDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_UNLOAD_PUNCHDDNC_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y. Some types of changes require that the unloaded data be converted by Db2 Admin Tool before the data is loaded. This parameter controls the user provided template for the converted load control card for the unloaded data.

Values:
a non-blank value

a non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_CLONE_TEMPLATE_UNLOAD_UNLDDN_NAME

Specifies the user provided template name for the UNLDDN file of the UNLOAD utility.

Values:

A Db2 template name; 1 to 8 characters

Default:

CUUNL

UTIL_CLONE_TEMPLATE_UNLOAD_UNLDDN_USE

Specifies whether to use a user provided template for the UNLDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_UNLOAD_UNLDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

UTIL_CLONE_TEMPLATE_UNLOAD_UNLDDNC_NAME

Specifies the user provided template name for the Db2 Admin Tool converted version of the UNLDDN file of the UNLOAD utility. Some types of changes require that the unloaded data to be converted by Db2 Admin Tool before the data can be loaded. This parameter controls the user provided template for the converted data set for the unloaded data.

Values:

A Db2 template name; 1 to 8 characters

Default:

CUUNLC

UTIL_CLONE_TEMPLATE_UNLOAD_UNLDDNC_USE

Specifies whether to use a user provided template for the Db2 Admin Tool converted version of the UNLDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_UNLOAD_UNLDDNC_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y. Some types of changes requires the unloaded data to be converted by Db2 Admin Tool before it can be loaded. This parameter controls the user provided template for the converted data set for the unloaded data.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

UTIL_CLONE_TEMPLATE_WORKDDN1_NAME

Specifies the user-provided template name for the first name for WORKDDN.

Values:

A Db2 template name; 1 to 8 characters

Default:

CLNWORK1

UTIL_CLONE_TEMPLATE_WORKDDN1_USE

Specifies whether to use a user-provided template for the first WORKDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_WORKDDN1_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the**GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

UTIL_CLONE_TEMPLATE_WORKDDN2_NAME

Specifies the user-provided template name for the second name for WORKDDN.

Values:

A Db2 template name; 1 to 8 characters

Default:

CLNWORK2

UTIL_CLONE_TEMPLATE_WORKDDN2_USE

Specifies whether to use a user-provided template for the second WORKDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_CLONE_TEMPLATE_WORKDDN2_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

a non-blank value

a non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

UTIL_CLONE_TEMPLATE_XMLCOL_NAME

Specifies the user-provided template name for XML columns.

Values

A Db2 template name; 1 to 8 characters

Default:

CLNXMLC

The data set name cannot exceed 35 bytes and must be in PDS format. Do not specify a member name.

UTIL_CLONE_TEMPLATE_XMLCOL_USE

Specifies whether to use a user-provided template for templates related to XML columns. If a non-blank value is specified, the template name for XML columns is determined from the **UTIL_CLONE_TEMPLATE_XMLCOL_NAME** parameter. This parameter is only in effect if the **GENERATE_TEMPLATES** is set to Y.

Values:

a non-blank value

Default:

S

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** is set to Y, and the template exists in the ADBTEMPL file.

UTIL_TEMPLATE_COPYDDN1_NAME

Specifies the user provided template name for the first file of COPYDDN.

Values:

A Db2 template name

The Db2 template name can be 1 to 8 character in length.

Default:
COPY1

UTIL_TEMPLATE_COPYDDN1_USE
Specifies whether to use a user provided template for the first COPYDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_COPYDDN1_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_COPYDDN2_NAME
Specifies the user provided template name for the second file of COPYDDN.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
COPY2

UTIL_TEMPLATE_COPYDDN2_USE
Specifies whether to use a user provided template for the second COPYDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_COPYDDN2_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_DISCARDDN_NAME
Specifies the user provided template name for the DISCARDDN file.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
DISC

UTIL_TEMPLATE_DISCARDDN_USE
Specifies whether to use a user provided template for the DISCARDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_DISCARDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y, and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_ERRDDN_DEVTYPE
Specifies whether the ERRDDN template is on a tape-like device, or on a DASD device.

Values:

TAPE
A removal media device, such as 3490 tape, or a 3490E tape drive.

DASD
A magnetic disk storage device, such as a direct access storage device (DASD).

Default:
DASD

UTIL_TEMPLATE_ERRDDN_NAME
Specifies the user provided template name for the ERRDDN file.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
ERR

UTIL_TEMPLATE_ERRDDN_USE
Specifies whether to use a user provided template for the ERRDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_ERRDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_FCCOPYDDN_NAME
Specifies the user provided template name for the FCCOPYDDN file.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
FCOPY

UTIL_TEMPLATE_FCCOPYDDN_USE
Specifies whether to use a user provided template for the FCCOPYDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_FCCOPYDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_LOBCOL_NAME
Specifies the user provided template name for LOB columns.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
LOBC

UTIL_TEMPLATE_LOBCOL_USE

Specifies whether to use a user provided template for templates related to LOB columns. If a non-blank value is specified, the template name for LOB columns is determined from the **UTIL_TEMPLATE_LOBCOL_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

UTIL_TEMPLATE_MAPDDN_DEVTYPE

Specifies whether the MAPDDN template is on a tape-like device, or on a DASD device.

Values:

TAPE

A removal media device, such as 3490 tape, or a 3490E tape drive.

DASD

A magnetic disk storage device, such as a direct access storage device (DASD).

Default:

DASD

UTIL_TEMPLATE_MAPDDN_NAME

Specifies the user provided template name for MAPDDN.

Values:

A Db2 template name

The Db2 template name can be 1 to 8 character in length.

Default:

MAP

UTIL_TEMPLATE_MAPDDN_USE

Specifies whether to use a user-provided template for the MAPDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_MAPDDN_NAME** parameter. This parameter is only in effect if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value.

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y, and the template exists in the ADBTEMPL file.

Default:

S

UTIL_TEMPLATE_PUNCHDDN_NAME

Specifies the user provided template name for the PUNCHDDN file of the REORG utility.

Values:

A Db2 template name

The Db2 template name can be 1 to 8 character in length.

Default:

PUNCH

UTIL_TEMPLATE_PUNCHDDN_USE

Specifies whether to use a user provided template for the PUNCHDDN file of the REORG utility. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_PUNCHDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_RECOVERYDDN1_NAME
Specifies the user-provided template name for the first file of RECOVERYDDN.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
RCVR1

UTIL_TEMPLATE_RECOVERYDDN1_USE
Specifies whether to use a user-provided template for the first RECOVERYDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_RECOVERYDDN1_NAME** parameter. This parameter is only in effect if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value.
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y, and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_RECOVERYDDN2_NAME
Specifies the user-provided template name for the second file of RECOVERYDDN.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
RCVR2

UTIL_TEMPLATE_RECOVERYDDN2_USE
Specifies whether to use a user-provided template for the second RECOVERYDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_RECOVERYDDN2_NAME** parameter. This parameter is only in effect if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value.
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y, and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_UNLDDN_NAME
Specifies the user provided template name for the UNLDDN file of the REORG utility.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
UNL

UTIL_TEMPLATE_UNLDDN_USE
Specifies whether to use a user provided template for the UNLDDN file of the REORG utility. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_UNLDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_UNLOAD_PUNCHDDN_NAME
Specifies the user provided template name for the PUNCHDDN file of the UNLOAD utility.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
UPUNCH

UTIL_TEMPLATE_UNLOAD_PUNCHDDN_USE
Specifies whether to use a user provided template for the PUNCHDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_UNLOAD_PUNCHDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_UNLOAD_PUNCHDDNC_NAME
Specifies the user provided template name for the Db2 Admin Tool converted version of the PUNCHDDN file of the UNLOAD utility. Some types of changes require that the unloaded data to be converted by Db2 Admin Tool before the data is loaded. This parameter controls the user provided template for the converted load control card for the unloaded data.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
UPUNCHC

UTIL_TEMPLATE_UNLOAD_PUNCHDDNC_USE
Specifies whether to use a user provided template for the PUNCHDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_UNLOAD_PUNCHDDNc_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y. Some types of changes requires the unloaded data to be converted by Db2 Admin Tool before it can be loaded. This parameter controls the user provided template for the converted load control card for the unloaded data.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_UNLOAD_UNLDDN_DEVTYPE
Specifies whether **UTIL_TEMPLATE_UNLOAD_UNLDDN_NAME** is on removable media or on a DASD device.

Values:

TAPE
A removal media device, such as 3490 tape, or a 3490E tape drive.

DASD
A magnetic disk storage device, such as a direct access storage device (DASD).

blank
The DEVTYPE option is not added. The Db2 default for this utility option is used.

Default:
DASD

UTIL_TEMPLATE_UNLOAD_UNLDDN_NAME
Specifies the user provided template name for the UNLDDN file of the UNLOAD utility.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
UUNL

UTIL_TEMPLATE_UNLOAD_UNLDDN_USE
Specifies whether to use a user provided template for the UNLDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_UNLOAD_UNLDDN_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_UNLOAD_UNLDDNC_NAME
Specifies the user provided template name for the Db2 Admin Tool converted version of the UNLDDN file of the UNLOAD utility. Some types of changes require that the unloaded data to be converted by Db2 Admin Tool before the data is loaded. This parameter controls the user provided template for the converted data set for the unloaded data.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
UUNLC

UTIL_TEMPLATE_UNLOAD_UNLDDNC_USE
Specifies whether to use a user provided template for the Db2 Admin Tool converted version of the UNLDDN file of the UNLOAD utility. If a non-blank value is specified, the template name is determined

from the **UTIL_TEMPLATE_UNLOAD_UNLDDNC_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y. Some types of changes require that the unloaded data to be converted by Db2 Admin Tool before the data is loaded. This parameter controls the user provided template for the converted data set for the unloaded data.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_WORKDDN1_DEVTYPE
Specifies whether the WORKDDN1 template is on a tape-like device, or on a DASD device.

Values:

TAPE
A removal media device, such as 3490 tape, or a 3490E tape drive.

DASD
A magnetic disk storage device, such as a direct access storage device (DASD).

Default:
DASD

UTIL_TEMPLATE_WORKDDN1_NAME
Specifies the user provided template name for the first name for WORKDDN.

Values:

A Db2 template name
The Db2 template name can be 1 to 8 character in length.

Default:
WORK1

UTIL_TEMPLATE_WORKDDN1_USE
Specifies whether to use a user provided template for the first WORKDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_WORKDDN1_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value
A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:
S

UTIL_TEMPLATE_WORKDDN2_DEVTYPE
Specifies whether the WORKDDN2 template is on a tape-like device, or on a DASD device.

Values:

TAPE
A removal media device, such as 3490 tape, or a 3490E tape drive.

DASD
A magnetic disk storage device, such as a direct access storage device (DASD).

Default:
DASD

UTIL_TEMPLATE_WORKDDN2_NAME
Specifies the user provided template name for the second name for WORKDDN.

Values:

A Db2 template name

The Db2 template name can be 1 to 8 character in length.

Default:

WORK2

UTIL_TEMPLATE_WORKDDN2_USE

Specifies whether to use a user provided template for the second WORKDDN file. If a non-blank value is specified, the template name is determined from the **UTIL_TEMPLATE_WORKDDN2_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

UTIL_TEMPLATE_XMLCOL_NAME

Specifies the user provided template name for XML columns.

Values:

A Db2 template name

The Db2 template name can be 1 to 8 character in length.

Default:

XMLC

UTIL_TEMPLATE_XMLCOL_USE

Specifies whether to use a user provided template for templates related to XML columns. If a non-blank value is specified, the template name for XML columns is determined from the **UTIL_TEMPLATE_XMLCOL_NAME** parameter. This parameter is in effect only if the **GENERATE_TEMPLATES** parameter is set to Y.

Values:

A non-blank value

A non-blank value indicates that the template name is used if the **GENERATE_TEMPLATES** parameter is set to Y and the template exists in the ADBTEMPL file.

Default:

S

VALIDATE_DDL

Specifies whether to validate the DDL that is being imported into change management. When the input to be imported is a changes file and not DDL, the value of this parameter is forced to N (NO).

This parameter validates the input DDL directly against the catalog for issuing the Db2 PREPARE statement, checking whether objects exist, and so on. Prerequisite changes are not considered, so an informational message is issued when Y is specified. For example, if the table that is in the DDL does not exist in the catalog, the validation process ends with an error. If a prerequisite change is in the process of creating the table, it is not considered.

The VALIDATE_DDL parameter is an import optional parameter that is limited to validating the DDL. It is not applicable to validating the changes file. For example, if you run a compare and import the changes file, the VALIDATE_DDL parameter is not used, and a warning message that the value of the parameter is forced to N is issued.

Values:

Y

Validates the DDL and displays the report in the job output.

N
The DDL is not validated.

Default:
N

VALIDATE_STMTEXIT
Specifies the REXX exit name for a VALIDATE statement.

Values:
Valid data set names. 1- 8 characters.

Default:
blank

VALIDATE_WSL
Specifies whether to validate the WSL after it is created. If the change has prerequisites, this option is forced to NO.

Values:

Y
Validate the WSL and display the report in the job output.

N
Do not validate the WSL.

Default:
N

VALIDATION_STMTEXIT
Specifies a REXX exec that will be called to validate a Db2 statement.

Values:

A valid data set member name
The data set member name can be 1 to 8 characters or blank.

blank
No validation will be done.

Default:
No default value.

WORKLIST_NAME
Specifies the member name for a PDS dataset that will be used for the WSL generation or run.

Values:
A valid PDS member name. 1 - 8 characters.

Default:
blank

WORKLIST_NAME_CONV
Specifies the member name for a partitioned data set (PDS) that has or will have the readable WSL. This parameter requires a valid PDS member name. Valid names are 1 - 8 characters. If the WSL that you specify in the PDS_FOR_WSL parameter is not found when ACTION_CONVERT_TO_READ_WSL= 'y', error message ADB9972E is displayed.

Values:
Valid PDS member names. 1 - 8 characters.

Default:
If a work statement list (WSL) is being generated ("ACTION_GENERATE_WSL" on page 670 = 'Y'), the default is the change tag value that is used by this WSL, as specified in WORKLIST_NAME.

If a change is being analyzed or run, the default is the change tag value that is used by the change. This value depends on the value of "CHGTAG_TYPE" on page 691

Otherwise, the default is blank.

Utility option parameters

Use the following CM batch parameters to set options for various utilities.

Requirement: To specify values for utility options, you must set the "USE_UTILITY_OPTIONS" on page 729 parameter to Y. Otherwise, the default values are used.

Recommendations: The following values are some generic recommendations for the utility parameters:

```
UTIL_COPY_FULL = 'Y'
UTIL_COPY_PARALLEL = '5'
UTIL_COPY_SHRLEVEL = 'C'
UTIL_REORG_KEEPDICTIONARY = 'Y'
UTIL_REORG_LOG = 'N'
UTIL_REORG_LONGLOG = 'D'
UTIL_REORG_MAXRO = '25'
UTIL_REORG_NOSYSREC = 'Y'
UTIL_REORG_RETRY = '6'
UTIL_REORG_RETRY_DELAY = '120'
UTIL_REORG_SHRLEVEL = 'C'
UTIL_REORG_SORTDATA = 'Y'
UTIL_REORG_SORTDEVT = 'SYSDA'
UTIL_REORG_TIMEOUT = 'T'
UTIL_REORG_DEADLINE = 'CURRENT TIMESTAMP + 4 HOURS'
UTIL_RUNSTATS_HISTORY = 'A'
UTIL_RUNSTATS_SHRLEVEL = 'C'
UTIL_RUNSTATS_UPDATE = 'A'
```

UTIL_CHECK_AUXERROR

Specifies the AUXERROR option for generated CHECK DATA utility statements. AUXERROR specifies the action that CHECK DATA is to take for LOB or XML column check errors.

Values:

R
AUXERROR REPORT is added.

I
AUXERROR INVALIDATE is added.

blank
The AUXERROR option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the CHECK DATA control statement (Db2 12 for z/OS)

UTIL_CHECK_DRAIN_WAIT

Specifies the DRAIN_WAIT option for generated CHECK DATA utility statements. DRAIN_WAIT specifies the number of seconds that the utility is to wait when it drains the table space or index.

Values:

A valid DRAIN_WAIT value for CHECK DATA; 1 - 1800
Specify a DRAIN_WAIT setting in the range 1 - 1800. The DRAIN_WAIT option is added with the specified value.

blank
The option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the CHECK DATA control statement (Db2 12 for z/OS)

UTIL_CHECK_EXCEPTIONS

Specifies the EXCEPTIONS option for generated CHECK DATA utility statements. EXCEPTIONS specifies the maximum number of exceptions, after which the utility terminates.

Values:

A valid EXCEPTIONS value for CHECK DATA)
Specify a valid EXCEPTIONS value in the range 0 - 32767. The EXCEPTIONS option is added with the specified value, for example: EXCEPTIONS 2

blank
The option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the CHECK DATA control statement (Db2 12 for z/OS)

UTIL_CHECK_INCLUDE_XML_TABLESPACES
Specifies the INCLUDE XML TABLESPACES option for generated CHECK DATA utility statements. INCLUDE XML TABLESPACES specifies whether the utility is to also check XML table spaces.

Values:

ALL
The INCLUDE XML TABLESPACES option is added.

blank
The INCLUDE XML TABLESPACES option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the CHECK DATA control statement (Db2 12 for z/OS)

UTIL_CHECK_RETRY
Specifies the RETRY option for generated CHECK DATA utility statements. RETRY specifies the maximum number of retries that the utility is to attempt.

Values:

A valid RETRY value for CHECK DATA
Specify a RETRY value in the range 0 - 255. The RETRY option is added with the specified value.

blank
The option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the CHECK DATA control statement (Db2 12 for z/OS)

UTIL_CHECK_RETRY_DELAY
Specifies the RETRY_DELAY option for generated CHECK DATA utility statements. RETRY_DELAY specifies the minimum duration, in seconds, between retries.

Values:

A valid RETRY_DELAY value for CHECK DATA
Specify a RETRY_DELAY setting in the range 1 - 1800. The RETRY_DELAY option is added with the specified value.

blank
The option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the CHECK DATA control statement (Db2 12 for z/OS)

UTIL_CHECK_SCOPE

Specifies the SCOPE option for generated CHECK DATA utility statements. SCOPE limits the scope of what the utility checks.

Values:

P
SCOPE PENDING is added.

X
SCOPE AUXONLY is added.

A
SCOPE ALL is added.

R
SCOPE REFONLY is added.

M
SCOPE XMLSCHEMAONLY is added.

blank
The SCOPE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the CHECK DATA control statement (Db2 12 for z/OS)

UTIL_CHECK_SORTDEVT

Specifies the SORTDEVT option for generated CHECK DATA utility statements. SORTDEVT specifies the device type for temporary data sets that are to be dynamically allocated by a sort program.

Values:

A valid SORTDEVT value for CHECK DATA
The SORTDEVT option is added with the specified value. For example: `SORTDEVT  SYSDA`.

Default:
SPACE_UNIT_NAME (See "SPACE_UNIT_NAME" on page 721.)

Related information:

Syntax and options of the CHECK DATA control statement (Db2 12 for z/OS)

UTIL_CHECK_SORTNUM

Specifies the SORTNUM option for generated CHECK DATA utility statements. SORTNUM specifies the number of temporary data sets that are to be dynamically allocated by the sort program.

Values:

A valid SORTNUM value for CHECK DATA.
Specify a SORTNUM value in the range 1 - 255. The SORTNUM option is added with the specified value.

Default:
4

Related information:

Syntax and options of the CHECK DATA control statement (Db2 12 for z/OS)

UTIL_CHECK_XMLSCHEMA

Specifies the XMLSCHEMA attribute of the INCLUDE XML TABLESPACES option for generated CHECK DATA utility statements. XMLSCHEMA specifies that the utility checks XML documents against the stored XML schema.

Values:

YES
The XMLSCHEMA option is added if the INCLUDE XML TABLESPACES option is also added.

NO
The XMLSCHEMA option is not added.

Default:
NO

Related information:

Syntax and options of the CHECK DATA control statement (Db2 12 for z/OS)

UTIL_COPY_CHANGELIMIT
Specifies the CHANGELIMIT option for generated COPY utility statements. CHANGELIMIT specifies the limit of changed pages at which an image copy is to be taken.

Values:

Y
The CHANGELIMIT option is added with the values that are specified by the UTIL_COPY_CHANGELIMIT_PERCENT_VALUE1 and UTIL_COPY_CHANGELIMIT_PERCENT_VALUE2 parameters.

A
The CHANGELIMIT(ANY) option is added.

blank
The CHANGELIMIT option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_COPY_CHANGELIMIT_PERCENT_VALUE1
Specifies the first value of the CHANGELIMIT option for generated COPY utility statements.

Values:

A percent value allowed by Db2
This value is specified with the CHANGELIMIT option.

blank
No value is specified with the CHANGELIMIT option. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_COPY_CHANGELIMIT_PERCENT_VALUE2
Specifies the second value of the CHANGELIMIT option for generated COPY utility statements.

Values:

A percent value allowed by Db2
This value is specified as the second value of the CHANGELIMIT option.

blank
A second value is not specified with the CHANGELIMIT option. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_COPY_CHANGELIMIT_REPORTONLY

Specifies the CHANGELIMIT REPORTONLY option for generated COPY utility statements. REPORTONLY specifies that image copy information is only displayed; no image copies are taken.

Values:

Y

The REPORTONLY option is added.

N

The REPORTONLY option is not added.

Default:

N

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_COPY_CHECKPAGE

Specifies the CHECKPAGE option for generated COPY utility statements. CHECKPAGE specifies that each page is to be checked for validity.

Values:

Y

The CHECKPAGE option is added.

N

The CHECKPAGE option is not added. The Db2 default for this utility option is used.

Default:

N

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_COPY_CONCURRENT

Specifies the CONCURRENT option for generated COPY utility statements. CONCURRENT specifies that DFSMSdss concurrent copy is to make the full image copy.

Values:

Y

The CONCURRENT option is added.

N

The CONCURRENT option is not added.

Default:

N

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_COPY_FLASHCOPY

Specifies the FLASHCOPY option for generated COPY utility statements. FLASHCOPY specifies whether FlashCopy® technology is used to create the copy.

Values:

Y

The FLASHCOPY YES option is added.

N

The FLASHCOPY NO option is not added.

C
The FLASHCOPY CONSISTENT option is added.

blank
The FLASHCOPY option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_COPY_FULL
Specifies the FULL option for generated COPY utility statements. FULL specifies whether the utility is to make a full or incremental image copy.

Values:

Y
The FULL YES option is added.

N
The FULL NO option is added.

blank
The FULL option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_COPY_NOCHECKPAGE
Specifies the NOCHECKPAGE option for generated COPY utility statements. NOCHECKPAGE indicates that COPY is to perform only basic checks on each page.

Values:

Y or YES
The NOCHECKPAGE option is added.

N or NO
The NOCHECKPAGE option is not added.

blank
The NOCHECKPAGE option is not added. The Db2 default for this utility option is used.

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

Default:
blank

UTIL_COPY_PARALLEL
Specifies the PARALLEL option for generated COPY utility statements. PARALLEL specifies the maximum number of objects in a list that are to be processed in parallel.

Values:

0 to 99999
The PARALLEL option is added as PARALLEL **`UTIL_COPY_PARALLEL`**. Where **`UTIL_COPY_PARALLEL`** is the value specified for this parameter.

blank
The PARALLEL option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_COPY_PARALLEL_TAPEUNITS

Specifies the PARALLEL TAPEUNITS option for generated COPY utility statements. TAPEUNITS specifies the maximum number of tape drives that the utility dynamically allocates for the list of objects to be processed in parallel.

Values:

0 to 32767

If the PARALLEL option is added, the TAPEUNITS *n* option is added. Where *n* is the value of this parameter.

blank

The TAPEUNITS option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_COPY_SHRLEVEL

Specifies the SHRLEVEL option for generated COPY utility statements. SHRLEVEL specifies whether other programs can access or update the table space or index while COPY is running.

Values:

C

The SHRLEVEL CHANGE option is added.

R

The SHRLEVEL REFERENCE option is added.

blank

The SHRLEVEL option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the COPY control statement (Db2 12 for z/OS)

UTIL_LISTDEF_PARTLEVEL

Specifies whether to generate the LISTDEF utility statement with the PARTLEVEL option. PARTLEVEL specifies the partition granularity. UTIL_LISTDEF_PARTLEVEL applies only when using the COPY utility to create image copies of partitioned table spaces.

Values:

Y

The LISTDEF *listdef-name*... PARTLEVEL statement is generated.

N

The LISTDEF *listdef-name*... PARTLEVEL statement is not generated.

Default:

N

Related information:

Syntax and options of the LISTDEF control statement (Db2 12 for z/OS)

UTIL_LOAD_BACKOUT

Specifies the BACKOUT option for generated LOAD utility statements. BACKOUT specifies whether to delete all rows loaded by the current LOAD operation if any record would leave the object unavailable.

Values:

YES or Y
The BACKOUT option is added.

NO or N
The BACKOUT option is not added.

blank
The BACKOUT option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_DECFLOAT_ROUNDMODE
Specifies the DECFLOAT_ROUNDMODE option for generated LOAD utility statements. DECFLOAT_ROUNDMODE specifies the rounding mode to use when DECFLOATs are manipulated.

Values:

C
The DECFLOAT_ROUNDMODE ROUND_CEILING option is added.

D
The DECFLOAT_ROUNDMODE ROUND_DOWN option is added.

F
The DECFLOAT_ROUNDMODE ROUND_FLOOR option is added.

HD
The DECFLOAT_ROUNDMODE ROUND_HALF_DOWN option is added.

HE
The DECFLOAT_ROUNDMODE ROUND_HALF_EVEN option is added.

HU
The DECFLOAT_ROUNDMODE ROUND_HALF_UP option is added.

U
The DECFLOAT_ROUNDMODE ROUND_UP option is added.

blank
The DECFLOAT_ROUNDMODE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_DEFINEAUX
Specifies the DEFINEAUX option for generated LOAD utility statements. DEFINEAUX specifies whether LOAD is to define all target auxiliary objects with the DEFINE NO attribute, regardless of whether data is to be loaded into these objects.

Values:

YES or Y
The DEFINEAUX YES option is added. LOAD defines all target LOB and XML objects and their indexes at the start of the utility execution.

NO or N
The DEFINEAUX NO option is added. LOAD takes no special action to define the target auxiliary objects.

blank
The DEFINEAUX option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_DISCARDS
Specifies the DISCARDS option for generated LOAD utility statements. DISCARDS specifies the maximum number of source records that are to be written on the discard data set.

Values:
A valid number in the range 0-2147483647

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_ENFORCE
Specifies the ENFORCE option for generated LOAD utility statements. ENFORCE specifies whether LOAD is to enforce check constraints and referential constraints, except informational referential constraints.

Values:

YES
The ENFORCE CONSTRAINTS option is added.

NO
The ENFORCE NO option is added.

Default:
NO

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_FLASHCOPY
Specifies the FLASHCOPY option for generated LOAD utility statements. FLASHCOPY specifies whether FlashCopy® technology is used.

Values:

Y
The FLASHCOPY YES option is added.

N
The FLASHCOPY NO option is added.

C
The FLASHCOPY CONSISTENT option is added.

blank
The FLASHCOPY option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_FORCE
Specifies the FORCE option for generated LOAD utility statements. FORCE specifies the action to be taken when the utility drains the table space.

Values:

N
The FORCE NONE option is added. No action is taken.

R
The FORCE READERS option is added. Read claimers are canceled when LOAD requests DRAIN ALL on the last drain retry.

A
The FORCE ALL option is added. Both read and write claimers are canceled when LOAD requests DRAIN ALL or DRAIN WRITERS on the last drain retry.

blank
The FORCE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_KEEPDICTIONARY
Specifies the KEEPDICTIONARY option for generated LOAD utility statements. KEEPDICTIONARY prevents the LOAD utility from building a new compression dictionary.

Values:

YES
The KEEPDICTIONARY option is added.

NO
The KEEPDICTIONARY option is not added. The Db2 default for this utility option is used.

Default:
NO

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_INDEXDEFER
Specifies the INDEXDEFER option for generated LOAD utility statements. INDEXDEFER specifies whether index builds are done during the BUILD phase of LOAD or are deferred until REBUILD INDEX is run manually.

Values:

NPI or N
The INDEXDEFER NPI option is added. The building of nonpartitioned indexes is not done as part of a BUILD phase of the LOAD utility.

ALL or A
The INDEXDEFER ALL option is added. No indexes are built as part of a BUILD phase of the LOAD utility.

NONE or NO
The INDEXDEFER NONE option is added. Indexes are built during the BUILD phase of LOAD.

blank
The INDEXDEFER option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_INDEXDEFER_NONUNIQUE

Specifies whether the INDEXDEFER NONUNIQUE option is added to generated LOAD utility statements. INDEXDEFER NONUNIQUE specifies that building of only nonunique indexes is deferred.

Values:

YES or Y

The INDEXDEFER NONUNIQUE option is added.

NO or N

The INDEXDEFER NONUNIQUE option is not added.

blank

The INDEXDEFER NONUNIQUE option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_LOG

Specifies the LOG option for generated LOAD utility statements. LOG specifies whether logging occurs during the load process.

Values:

YES

LOG YES is added.

NO

LOG NO is added.

NOC

LOG NO NOCOPYPEND is added.

blank

The LOG option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_NOCHECKPEND

Specifies the NOCHECKPEND option for generated LOAD utility statements. NOCHECKPEND specifies that LOAD does not set the target table space in the CHECK-pending status if at least one referential constraint or check constraint is defined for the table.

Values:

YES or Y

The NOCHECKPEND option is added.

NO or N

The NOCHECKPEND option is not added.

blank

The NOCHECKPEND option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_NOSUBS

Specifies the NOSUBS option for generated LOAD utility statements. NOSUBS specifies that LOAD is not to accept substitution characters in a string.

Values:

YES or Y
The NOSUBS option is added.

NO or N
The NOSUBS option is not added.

blank
The NOSUBS option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_PARALLEL

Specifies the maximum number of subtasks that are to be used in parallel when loading a table space.

Values:

YES
The PARALLEL option is added.

integer
0-32767. The PARALLEL option is added to the utility statement with the specified value.

blank
The PARALLEL option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_PREFORMAT

Specifies the PREFORMAT option for generated LOAD utility statements. PREFORMAT specifies that the remaining pages are to be preformatted up to the high-allocated RBA in the partition and its corresponding partitioning index space

Values:

YES or Y
The PREFORMAT option is added.

NO or N
The PREFORMAT option is not added.

blank
The PREFORMAT option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_PRESORT

Specifies the PRESORT option for generated LOAD utility statements. PRESORT specifies that input records are to be sorted in clustering order before loading them into the target table space. Existing rows in the table space are not affected.

Values:

YES or Y
The PRESORT option is added.

NO or N
The PRESORT option is not added.

blank
The PRESORT option is not added.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_PRESORTED
Specifies the PRESORTED option for generated LOAD utility statements. PRESORTED specifies whether the input data set has already been sorted in clustering key order.

Values:

YES or Y
The PRESORTED YES option is added. Input data is sorted.

NO or N
The PRESORTED NO option is added. The input data set is not sorted.

blank
The PRESORTED option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_REPLACE
Specifies the REPLACE option for generated LOAD utility statements. REPLACE specifies whether the table space and all its indexes need to be reset to empty before records are loaded.

Values:

YES
The REPLACE option is added.

NO
The REPLACE option is not added.

blank
The REPLACE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_RESUME
Specifies the RESUME option for generated LOAD utility statements. RESUME specifies whether records are to be loaded into an empty or non-empty table space.

Values:

YES
RESUME YES is added.

NO
RESUME NO is added.

blank
The RESUME option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_REUSE
Specifies the REUSE option for generated LOAD utility statements. REUSE specifies whether LOAD reuses Db2-managed data sets.

Values:

YES
The REUSE option is added.

NO
The REUSE option is not added.

Default:
NO

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_SHRLEVEL
Specifies the SHRLEVEL option for generated LOAD utility statements. SHRLEVEL specifies the extent to which applications can concurrently access the table space or partition during the LOAD utility job.

Important: UTIL_LOAD_SHRLEVEL is currently ignored by CM batch. If you set UTIL_LOAD_SHRLEVEL="C", warning ADB9985W is issued and the Db2 default SHRLEVEL value will be used.

Values:

N
The SHRLEVEL NONE option is added.

C
The SHRLEVEL CHANGE option is added.

blank
The SHRLEVEL option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_SORTDEVT
Specifies the SORTDEVT option for generated LOAD utility statements. SORTDEVT specifies the device type for temporary data sets that are to be dynamically allocated by the external sort program

Values:

A valid SORTDEVT value for LOAD
The SORTDEVT option is added with the specified value. For example: `SORTDEVT SYSDA`..

SPACE_UNIT_NAME
See SPACE_UNIT_NAME

Default:
SPACE_UNIT_NAME

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_SORTKEYS

Specifies the SORTKEYS option for generated LOAD utility statements. SORTKEYS specifies the number of index keys that are to be sorted.

Values:

A valid SORTKEYS value for LOAD.
The SORTKEYS option is added with the specified value.

0
The SORTKEYS option is not added. The Db2 default for this utility option is used.

Default:
0

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_SORTNUM

Specifies the SORTNUM option for generated LOAD utility statements. SORTNUM specifies the number of temporary data sets that are to be dynamically allocated by the sort application program

Values:

A valid SORTNUM value for LOAD.
The SORTNUM option is added with the specified value.

Default:
8

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_STATISTICS

Specifies whether the STATISTICS option is to be added to generated LOAD utility statements. STATISTICS specifies that the utility is to gather inline statistics.

Values:

Y or YES
The STATISTICS option is added.

N or NO
The STATISTICS option is not added.

blank
The STATISTICS option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_STATISTICS_FORCEROLLUP

Specifies the FORCEROLLUP option for generated LOAD utility statements. FORCEROLLUP specifies whether aggregation or rollup of statistics is to take place when RUNSTATS is executed even if statistics have not been gathered on some partitions.

Values:

Y or YES
The FORCEROLLUP YES option is added.

N or NO
The FORCEROLLUP NO option is added.

blank
The FORCEROLLUP option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_STATISTICS_HISTORY
Specifies the HISTORY option for generated LOAD utility statements. HISTORY specifies that all catalog table inserts or updates to the catalog history tables are to be recorded.

Values:

A
The HISTORY ALL option is added. ALL indicates that all collected statistics are to be updated in the catalog history tables.

P
The HISTORY ACCESSPATH option is added. ACCESSAPTH indicates that only the catalog history table columns that provide statistics that are used for access path selection are to be updated.

S
The HISTORY SPACE option is added. SPACE indicates that only space-related catalog statistics are to be updated in catalog history tables.

N
The HISTORY NONE option is added. NONE indicates that no catalog history tables are to be updated with the collected statistics.

blank
The HISTORY option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_STATISTICS_INVALIDATECACHE
Specifies the INVALIDATECACHE option for generated LOAD utility statements. INVALIDATECACHE indicates whether statements in the dynamic statement cache are invalidated as a result of the inline statistics collection.

Values:

Y or YES
The INVALIDATECACHE YES option is added.

N or NO
The INVALIDATECACHE NO option is added.

blank
The INVALIDATECACHE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_STATISTICS_REPORT
Specifies the REPORT option for generated LOAD utility statements. REPORT specifies whether a set of messages is to be generated to report the collected statistics.

Values:

Y or YES
The REPORT YES option is added.

N or NO
The REPORT NO option is added.

blank
The REPORT option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_STATISTICS_TABLE_SAMPLE
Specifies the TABLE SAMPLE option for generated LOAD utility statements. TABLE SAMPLE indicates the percentage of rows to be sampled when collecting statistics on non-leading-indexed columns of an index or non-indexed columns.

Values:

integer
The TABLE SAMPLE *integer* option is added. *integer* can be a value in the range 1 - 100.

blank
The TABLE SAMPLE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_STATISTICS_UPDATE
Specifies the UPDATE option for generated LOAD utility statements. UPDATE indicates whether the collected statistics are to be inserted into the catalog tables.

Values:

A
The UPDATE ALL option is added. ALL indicates that all collected statistics are to be updated in the catalog.

P
The UPDATE ACCESSPATH option is added. ACCESSAPTH indicates that only the catalog table columns that provide statistics that are used for access path selection are to be updated.

S
The UPDATE SPACE option is added. SPACE indicates that only the catalog table columns that provide statistics to help database administrators assess the status of a particular table space or index are to be updated.

N
The UPDATE NONE option is added. NONE indicates that no catalog tables are to be updated with the collected statistics.

blank
The UPDATE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_LOAD_STATISTICS_USE_PROFILE

Specifies whether the USE PROFILE option is added to generated LOAD utility statements. USE PROFILE specifies that a stored statistics profile is used to gather statistics for a table.

Values:

Y or YES
The USE PROFILE option is added.

N or NO
The USE PROFILE option is not added.

blank
The USE PROFILE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

UTIL_REBUILD_SHRLEVEL

Specifies the SHRLEVEL option for generated REBUILD INDEX utility statements. SHRLEVEL specifies the type of access that is to be allowed for the object that is being processed by the utility.

Values:

C
The SHRLEVEL CHANGE option is added.

R
The SHRLEVEL REFERENCE option is added.

blank
The SHRLEVEL option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REBUILD INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_AUX

Specifies the AUX option for generated REORG TABLESPACE utility statements. AUX specifies that associated LOB table spaces are also to be reorganized.

Values:

YES
AUX YES is added.

NO
AUX NO is added.

blank
The AUX option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_DEADLINE

Specifies the DEADLINE option for generated REORG TABLESPACE utility statements. DEADLINE specifies the deadline for the SWITCH phase to begin.

Values:

N
DEADLINE NONE is added.

timestamp
DEADLINE *timestamp* is added.

labeled-duration-expression
DEADLINE *labeled-duration-expression* is added.

blank
The DEADLINE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_DELAY
Specifies the DELAY option for generated REORG TABLESPACE utility statements. DELAY specifies the minimum delay before performing the action that is specified by the LONGLOG parameter.

Values:
integer, blank

integer
The DELAY option is added to the utility statement with the specified value. *integer* is the number of seconds.

blank
The DELAY option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_DRAIN
Specifies the DRAIN option for generated REORG TABLESPACE utility statements. DRAIN specifies drain behavior.

Values:

W
The DRAIN WRITERS option is added to the utility statement.

A
The DRAIN ALL option is added to the utility statement.

blank
The DRAIN ALL option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_DRAIN_ALLPARTS
Specifies the DRAIN_ALLPARTS option for generated REORG TABLESPACE utility statements. DRAIN_ALLPARTS specifies drain behavior during a partition-level REORG operation when a nonpartitioned secondary index is defined on a partitioned table space.

Values:

YES
The DRAIN_ALLPARTS YES option is added to the utility statement.

NO
The DRAIN_ALLPARTS NO option is added to the utility statement.

blank
The DRAIN_ALLPARTS option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_DRAIN_WAIT
Specifies the DRAIN_WAIT option for generated REORG TABLESPACE utility statements. DRAIN_WAIT specifies hows long the utility waits when draining the table space or index.

Values:

integer
A valid DRAIN_WAIT value for REORG. The DRAIN_WAIT option is added with the specified value.

blank
The DRAIN ALL option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_DROP_PART
Specifies the DROP_PART option for generated REORG TABLESPACE utility statements. DROP_PART controls whether empty partitions are deleted.

Values:

YES
DROP_PART YES is added.

NO
DROP_PART NO is added.

blank
The DROP_PART option is not added; The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_FASTSWITCH
Specifies the FASTSWITCH option for generated REORG TABLESPACE utility statement. FASTSWITCH specifies which switch methodology is to use for the reorganization.

Values:

Y
FASTSWITCH YES is added.

N
FASTSWITCH NO is added.

blank
The FASTSWITCH option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_FLASHCOPY
Specifies the FLASHCOPY option for generated REORG TABLESPACE utility statements. FLASHCOPY specifies whether FlashCopy® technology is used.

Values:

Y
FLASHCOPY YES is added.

C
FLASHCOPY CONSISTENT is added.

N
FLASHCOPY NO is added.

blank
The FLASHCOPY option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_FORCE
Specifies the FORCE option for generated REORG TABLESPACE utility statements. FORCE specifies the action to be taken when the utility drains the table space.

Values:

N
The FORCE NONE option is added. No action is taken.

R
The FORCE READERS option is added. Read claimers are canceled when REORG requests DRAIN ALL on the last drain retry.

A
The FORCE ALL option is added. Both read and write claimers are canceled when REORG requests DRAIN ALL or DRAIN WRITERS on the last drain retry.

blank
The FORCE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_ICLIMIT_DASD
Specifies the ICLIMIT_DASD option for generated REORG TABLESPACE utility statements. ICLIMIT_DASD specifies the maximum number of sequential image copies that REORG TABLESPACE can allocate to DASD.

Values:

integer
An integer value in the range 0 - 32767.

blank
The ICLIMIT_DASD option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_ICLIMIT_TAPE
Specifies the ICLIMIT_TAPE option for generated REORG TABLESPACE utility statements. ICLIMIT_TAPE specifies the maximum number of sequential image copies that REORG TABLESPACE can allocate to tape.

Values:

integer
An integer value in the range 0 - 32767.

blank
The ICLIMIT_TAPE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_CLONE
Specifies the CLONE option for generated REORG INDEX utility statements. CLONE specifies that only index spaces and indexes on clone tables are to be reorganized.

Values:

Y
CLONE is added.

N
CLONE is not is added.

Default:
N

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_DRAIN
Specifies the DRAIN option for generated REORG INDEX utility statements. DRAIN specifies drain behavior.

Values:

W
The DRAIN WRITERS option is added to the utility statement.

A
The DRAIN ALL option is added to the utility statement.

blank
The DRAIN ALL option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_FASTSWITCH

Specifies the FASTSWITCH option for generated REORG INDEX utility statements. FASTSWITCH specifies which switch methodology is to use for the reorganization.

Values:

Y

FASTSWITCH YES is added.

N

FASTSWITCH NO is added.

blank

The FASTSWITCH option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_FLASHCOPY

Specifies the FLASHCOPY option for generated REORG INDEX utility statements. FLASHCOPY specifies whether FlashCopy® technology is used.

Values:

Y

FLASHCOPY YES is added.

C

FLASHCOPY CONSISTENT is added.

N

FLASHCOPY NO is added.

blank

The FLASHCOPY option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_FORCE

Specifies the FORCE option for generated REORG INDEX utility statements. FORCE specifies the action to be taken when the utility drains the table space.

Values:

A

The FORCE ALL option is added. Both read and write claimers are canceled.

R

The FORCE READERS option is added. Read claimers are canceled.

N

The FORCE NONE option is added. No action is taken.

blank

The FORCE option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_LASTLOG

Specifies the LASTLOG option for generated REORG INDEX utility statements with the SHRLEVEL CHANGE option. LASTLOG controls whether the utility applies log records during the last log iteration in the LOG phase.

To use this parameter, the UTIL_REORG_INDEX_ SHRLEVEL parameter value must be C. Otherwise, UTIL_REORG_INDEX_LASTLOG is ignored.

Values:

YES or Y
The LASTLOG YES option is added.

NO or N
The LASTLOG NO option is added. If you specify NO or N, UTIL_REORG_INDEX_DRAIN must be set to A.

blank
The LASTLOG option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

s

UTIL_REORG_INDEX_LEAFDISTLIMIT

Specifies the LEAFDISTLIMIT option for generated REORG INDEX utility statements.

Values:
A valid LEAFDISTLIMIT value for REORG INDEX, blank

A valid LEAFDISTLIMIT value for REORG INDEX
LEAFDISTLIMIT is added with the specified value.

blank
The LEAFDISTLIMIT option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_NOSYSUT1

Specifies the NOSYSUT1 option for generated REORG INDEX utility statements. NOSYSUT1 specifies that REORG INDEX is not to use the SYSUT1 work data set to hold the unloaded index keys but instead pass these keys in memory. Note that Db2 ignores this keyword if SHRLEVEL NONE is specified or no value is specified for SHRLEVEL, in which case SHRLEVEL NONE is used by default.

Values:

Y
The NOSYSUT1 option is added.

N
The NOSYSUT1 option is not added.

blank
The NOSYSUT1 option is not added.

Default:
blank

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_PARALLEL

Specifies the PARALLEL option for generated REORG INDEX utility statements. PARALLEL specifies the maximum number of subtasks that are to be started in parallel to reorganize the index.

Values:

n

The PARALLEL *n* option is added. *n* must be an integer between 0 and 32767, inclusive.

blank

The PARALLEL option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_PREFORMAT

Specifies the PREFORMAT option for generated REORG INDEX utility statements. PREFORMAT specifies that the remaining pages are to be preformatted.

Values:

Y

PREFORMAT is added.

N

PREFORMAT is not added

Default:

N

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_REPORTONLY

Specifies the REPORTONLY option for generated REORG INDEX utility statements. REPORTONLY specifies the utility is to only recommend whether a reorganization is necessary; no reorganization is performed.

Values:

Y

REPORTONLY is added.

N

REPORTONLY is not added.

Default:

N

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_REUSE

Specifies the REUSE option for generated REORG INDEX utility statements. REUSE specifies that the utility to reuse Db2-managed data sets.

Values:

Y

REUSE is added

N

REUSE is not added.

Default:
blank

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_SHRLEVEL

Specifies the SHRLEVEL option for generated REORG INDEX utility statements. SHRLEVEL indicates the type of access that is allowed during REORG INDEX processing.

Values:

N
SHRLEVEL NONE is added.

C
SHRLEVEL CHANGE is added. However, the option might not be specified, or might be converted to SHRLEVEL REFERENCE for some generated REORG index statements. SHRLEVEL CHANGE is processed based on the Db2 SHRLEVEL CHANGE restrictions.

R
SHRLEVEL REFERENCE is added. However, the option might not be specified for some generated REORG index statements. SHRLEVEL REFERENCE is processed based on the Db2 SHRLEVEL REFERENCE restrictions.

blank
The SHRLEVEL option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_SORTDEVT

Specifies the SORTDEVT option for generated REORG INDEX utility statements. SORTDEVT specifies the device type for temporary data sets that are to be dynamically allocated by the external sort program

Values:

A valid SORTDEVT value for REORG INDEX
The SORTDEVT option is added with the specified value. For example: `SORTDEVT SYSDA`.

blank
The SORTDEVT option is not added. The Db2 default for this utility option is used.

Default:
SPACE_UNIT_NAME (See "SPACE_UNIT_NAME" on page 721.)

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDEX_SORTNUM

Specifies the SORTNUM option for generated REORG INDEX utility statements. SORTNUM specifies the number of temporary data sets that are to be dynamically allocated when collecting statistics for a data-partitioned secondary index.

Values:

A valid SORTNUM value for REORG INDEX
The SORTNUM option is added with the specified value.

blank
The SORTNUM option is not added. The Db2 default for this utility option is used.

Default:
4

Related information:

Syntax and options of the REORG INDEX control statement (Db2 12 for z/OS)

UTIL_REORG_INDREFLIMIT

Specifies the INDREFLIMIT option for generated REORG TABLESPACE utility statements.

Values:

A valid INDREFLIMIT value for REORG

INDREFLIMIT is added with the specified value.

blank

The INDREFLIMIT option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_KEEPDICTIONARY

Specifies the KEEPDICTIONARY option for generated REORG TABLESPACE utility statements. KEEPDICTIONARY prevents the utility from building a new compression dictionary when unloading the rows.

Values:

Y

KEEPDICTIONARY is added.

N

KEEPDICTIONARY is not added.

Default:

N

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_LASTLOG

Specifies the LASTLOG option for generated REORG TABLESPACE utility statements with the SHRLEVEL CHANGE option. LASTLOG controls whether the utility applies log records during the last log iteration in the LOG phase.

To use this parameter, the UTIL_REORG_SHRLEVEL parameter value must be C. Otherwise, UTIL_REORG_LASTLOG is ignored.

Values:

YES or Y

The LASTLOG YES option is added.

NO or N

The LASTLOG NO option is added. If you specify NO or N, UTIL_REORG_DRAIN must be set to A.

blank

The LASTLOG option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

s

UTIL_REORG_LISTPARTS

Specifies the LISTPARTS option for generated REORG TABLESPACE utility statements that use a LISTDEF list with PARTLEVL. LISTPARTS specifies the maximum number of data partitions to be reorganized in a single REORG job.

The **`UTIL_REORG_LISTPARTS`** and **`UTIL_REORG_PARALLEL`** parameters are mutually exclusive.

Values:

Positive integer

The LISTPARTS option is added with the specified value.

blank

The LISTPARTS option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_LOG

Specifies the LOG option for generated REORG TABLESPACE utility statements. LOG specifies whether records are to be logged.

Values:

Y

LOG YES is added.

N

LOG NO is added.

blank

The LOG option is not added. The Db2 default for this utility option is used.

Default:

N

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_LOGRANGES

Specifies the LOGRANGES option for generated REORG TABLESPACE utility statements. LOGRANGES specifies whether REORG is to use SYSLGRNX information for the LOG phase.

Values:

YES

LOGRANGES YES is added.

NO

LOGRANGES NO is added.

blank

The LOGRANGES option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_LONGLOG

Specifies the LONGLOG option for generated REORG TABLESPACE utility statements. LONGLOG specifies the action to take if the utility is not reading the application log quickly enough.

Values:

C
The LONGLOG CONTINUE option is added to the utility statement.

T
The LONGLOG TERM option is added to the utility statement.

D
The LONGLOG DRAIN option is added to the utility statement.

blank
The LONGLOG option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_MAPPINGDATABASE
Specifies the MAPPINGDATABASE option for generated REORG TABLESPACE utility statements. MAPPINGDATABASE specifies the database in which the utility implicitly creates the mapping table and index objects.

Values:
A database name; 1 to 8 characters.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_MAPTABLE_NAME
Specifies the mapping table name for generated REORG TABLESPACE utility statements.

Values:
Valid table owner name; 1 to 128 characters

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_MAPTABLE_OWNER
Specifies the mapping table owner for generated REORG utility statements.

Values:
Valid table owner name; 1 to 128 characters

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_MAXRO
Specifies the MAXRO option for generated REORG TABLESPACE utility statements. MAXRO specifies the maximum amount of time for the last iteration of log processing.

Values:

integer
The MAXRO option is added to the utility statement with the specified value.

D

The MAXRO DEFER option is added to the utility statement.

blank

The MAXRO option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_NEWMAXRO

Specifies the NEWMAXRO option for generated REORG TABLESPACE utility statements. NEWMAXRO specifies the maximum amount of time for the last log iteration after SWITCHTIME is met.

Values:

NONE

The NEWMAXRO NONE option is added.

integer

The NEWMAXRO option with the specified integer value is added.

blank

The NEWMAXRO option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_NOCHECKPEND

Specifies the NOCHECKPEND option for generated REORG TABLESPACE utility statements. NOCHECKPEND specifies that when REORG discards records from a parent table in at least one referential integrity relationship, the utility does not set CHECK-pending status on the dependent table spaces.

Values:

YES or Y

NOCHECKPEND is added.

NO or N

NOCHECKPEND is not added.

blank

NOCHECKPEND is not added.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_NOSYSREC

Specifies the NOSYSREC option for generated REORG TABLESPACE utility statements. NOSYSREC specifies that the utility is not to use an unload data set.

Values:

Y

The NOSYSREC option is added.

N

The NOSYSREC option is not added.

Default:
N

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_OFFPOSLIMIT
Specifies the OFFPOSLIMIT option for generated REORG TABLESPACE utility statements.

Values:

A valid OFFPOSLIMIT value for REORG
OFFPOSLIMIT is added with the specified value.

blank
The OFFPOSLIMIT option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_PARALLEL
Specifies the PARALLEL option for generated REORG TABLESPACE utility statements. PARALLEL specifies the maximum number of subtasks that are to be started in parallel to reorganize a table space.

Values:

YES
The PARALLEL option is added.

integer
A valid PARALLEL value for REORG. The PARALLEL option is added to the utility statement with the specified value.

blank
The PARALLEL option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_PREFORMAT
Specifies the PREFORMAT option for generated REORG TABLESPACE utility statements. PREFORMAT specifies that the remaining pages are to be preformatted.

Values:

Y
PREFORMAT is added.

N
PREFORMAT is not added.

Default:
N

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_RECLUSTER
Specifies the RECLUSTER option for generated REORG TABLESPACE utility statements. RECLUSTER specifies whether data records are to be reclustered.

Values:

YES
RECLUSTER YES is added.

NO
RECLUSTER NO is added.

blank
The RECLUSTER option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_RETRY
Specifies the RETRY option for generated REORG TABLESPACE utility statements. RETRY specifies the maximum number of retries that the utility is to attempt.

Values:

A valid RETRY value for REORG
The RETRY option is added with the specified value.

blank
The RETRY option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_RETRY_DELAY
Specifies the RETRY_DELAY option for generated REORG TABLESPACE utility statements. RETRY_DELAY specifies the minimum duration, in seconds, between retries.

Values:

A valid RETRY_DELAY value for REORG
The RETRY_DELAY option is added with the specified value.

blank
The RETRY_DELAY option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_REUSE
Specifies the REUSE option for generated REORG TABLESPACE utility statements. REUSE specifies that the utility is to logically reuse Db2-managed data sets.

Values:

Y
REUSE is added.

N
REUSE is not added.

Default:
N

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_SHRLEVEL

Specifies the SHRLEVEL option for generated REORG TABLESPACE utility statements. SHRLEVEL indicates the type of access that is to be allowed during utility processing.

Values:

N
SHRLEVEL NONE is added.

C
SHRLEVEL CHANGE is added. However, the option might not be specified, or might be converted to SHRLEVEL REFERENCE for some generated REORG table space statements. SHRLEVEL CHANGE is processed based on the Db2 SHRLEVEL CHANGE restrictions.

R
SHRLEVEL REFERENCE is added. However, the option might not be specified for some generated REORG table space statements. SHRLEVEL REFERENCE is processed based on the Db2 SHRLEVEL REFERENCE restrictions.

blank
Either SHRLEVEL CHANGE or SHRLEVEL REFERENCE is added; Db2 Admin Tool determines the best value (CHANGE or REFERENCE) depending on the circumstance.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_SORTDATA

Specifies the SORTDATA option for generated REORG TABLESPACE utility statements. SORTDATA specifies that the data is to be unloaded by a table space scan, and sorted in clustering order.

Values:

Y
SORTDATA is added.

N
SORTDATA NO is added.

blank
The SORTDATA option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_SORTDEVT

Specifies the SORTDEVT option for generated REORG TABLESPACE utility statements. SORTDEVT specifies the device type for temporary data sets that are to be dynamically allocated by the external sort program.

Values:

A valid SORTDEVT value for REORG
The SORTDEVT option is added with the specified value. For example: `SORTDEVT  SYSDA`.

blank
The SORTDEVT option is not added. The Db2 default for this utility option is used.

Default:
SPACE_UNIT_NAME (See "SPACE_UNIT_NAME" on page 721.)

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_SORTKEYS

Specifies the SORTKEYS option for generated REORG TABLESPACE utility statements.

Beginning in DB2 UDB for z/OS 8, the SORTKEYS behavior is the default. The SORTKEYS keyword is ignored.

Values:

Y

SORTKEYS is added.

N

SORTKEYS is not added.

Default:

N

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_SORTNPSI

Specifies the SORTNPSI option for generated REORG TABLESPACE utility statements. SORTNPSI specifies when REORG TABLESPACE PART is to sort all keys of a non-partitioned secondary index.

Values:

AUTO or A

SORTNPSI AUTO is added.

YES or Y

SORTNPSI YES is added.

NO or N

SORTNPSI NO is added.

blank

The SORTNPSI option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_SORTNUM

Specifies the SORTNUM option for generated REORG TABLESPACE utility statements. SORTNUM specifies the number of temporary data sets that are to be dynamically allocated for all sorts that the utility performs.

Values:

A valid SORTNUM value for REORG

The SORTNUM option is added with the specified value.

blank

The SORTNUM option is not added. The Db2 default for this utility option is used.

Default:

4

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS

Specifies the STATISTICS option for generated REORG TABLESPACE utility statements. STATISTICS specifies that statistics are to be gathered.

Values:

Y

The STATISTICS option is added.

N

The STATISTICS option is not added. Any other specified REORG statistics options are not used.

blank

The STATISTICS option is conditionally added. It is added if a REORG statistics option was explicitly specified. For example, if a value for SAMPLE was specified using the **`UTIL_REORG_STATISTICS_TABLE_SAMPLE`** parameter.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS_FORCEROLLUP

Specifies the FORCEROLLUP option for generated REORG TABLESPACE utility statements. FORCEROLLUP specifies whether statistics are to be aggregated.

Values:

Y

FORCEROLLUP YES is added.

N

FORCEROLLUP NO is added.

blank

The FORCEROLLUP option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS_HISTORY

Specifies the HISTORY option for generated REORG TABLESPACE utility statements. HISTORY specifies that all catalog table inserts or updates to the catalog history tables are to be recorded.

Values:

A

HISTORY ALL is added.

P

HISTORY ACCESSPATH is added.

S

HISTORY SPACE is added.

N

HISTORY NONE is added.

blank

The HISTORY option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS_INDEX_HISTOGRAM

Specifies the HISTOGRAM option for generated REORG TABLESPACE utility statements. HISTOGRAM specifies that histogram statistics are to be gathered for the specified group of columns.

Values:

Y
The HISTOGRAM option is added.

N
The HISTOGRAM option is not added. Any other specified HISTOGRAM options are not used.

blank
The HISTOGRAM option is conditionally added. It is added if a value is specified for the **`UTIL_REORG_STATISTICS_INDEX_NUMCOLS`** parameter or the **`UTIL_REORG_STATISTICS_INDEX_NUMQUANTILES`** parameter.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS_INDEX_NUMCOLS

Specifies the NUMCOLS option for generated REORG TABLESPACE utility statements. NUMCOLS specifies the number of key columns that are to be concatenated when collecting histogram statistics from the specified index.

If a value is not specified for the **`UTIL_REORG_STATISTICS_INDEX_NUMCOLS`** parameter but a value is specified for the **`UTIL_REORG_STATISTICS_INDEX_NUMQUANTILES`** parameter, NUMCOLS 1 is added to generated REORG statements.

Values:

1 - 64
The NUMCOLS option is added with the specified value.

blank
The NUMCOLS option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS_INDEX_NUMQUANTILES

Specifies the NUMQUANTILES option for generated REORG TABLESPACE utility statements. NUMQUANTILES specifies the number of quantiles that are requested.

Values:

1 - 100
The NUMQUANTILES option is added with the specified value.

blank
The NUMQUANTILES option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS_INVALIDATECACHE

Specifies the INVALIDATECACHE option for the generated REORG TABLESPACE utility statements with inline statistics. INVALIDATECACHE indicates whether statements in the dynamic statement cache are invalidated as a result of the inline statistics collection.

If you specify the **UTIL_REORG_STATISTICS_INVALIDATECACHE** parameter, you must also specify `UTIL_REORG_STATISTICS ='Y'`.

Values:

Y
INVALIDATECACHE YES is added to the utility statement.

N
INVALIDATECACHE NO is added to the utility statement.

blank
The INVALIDATECACHE option is not added to the utility statement. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS_REPORT

Specifies the REPORT option for generated REORG TABLESPACE utility statements. REPORT specifies whether a set of messages is to be generated to report the collected statistics.

Values:

Y
REPORT YES is added.

N
REPORT NO is added.

blank
The REPORT option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS_TABLE_SAMPLE

Specifies the SAMPLE option for generated REORG TABLESPACE utility statements. SAMPLE indicates the percentage of rows to be sampled when collecting statistics on non-leading-indexed columns of an index or non-indexed columns.

Values:

1 - 100
The SAMPLE option is added with the specified value.

blank
The SAMPLE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS_UPDATE

Specifies the UPDATE option for generated REORG TABLESPACE utility statements. UPDATE specifies whether the collected statistics are to be inserted into catalog tables.

Values:

A
UPDATE ALL is added.

P
UPDATE ACCESSPATH is added.

S
UPDATE SPACE is added.

N
UPDATE NONE is added.

blank
The UPDATE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_STATISTICS_USE_PROFILE

Specifies whether the USE PROFILE option is to be added to the generated REORG TABLESPACE utility statements. USE PROFILE specifies that a statistics profile is to be used to gather statistics.

Values:

Y
USE PROFILE is added.

N
USE PROFILE is not added.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)
Statistics profiles (Db2 12 for z/OS)

UTIL_REORG_SWITCHTIME

Specifies the SWITCHTIME option for generated REORG TABLESPACE utility statements. SWITCHTIME specifies the time for the final log iteration of the LOG phase to begin.

Values:

NONE
SWITCHTIME NONE is added.

timestamp
SWITCHTIME *timestamp* is added.

labeled-duration-expression
SWITCHTIME *labeled-duration-expression* is added.

blank
The SWITCHTIME option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_REORG_TIMEOUT

Specifies the TIMEOUT option for generated REORG TABLESPACE utility statements. TIMEOUT specifies the action that is to be taken if the REORG utility gets a timeout condition.

Values:

T

The TIMEOUT TERM option is added to the utility statement.

A

The TIMEOUT ABEND option is added to the utility statement.

blank

The TIMEOUT option is not added to the utility statement. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the REORG TABLESPACE control statement (Db2 12 for z/OS)

UTIL_RUNSTATS_HISTORY

Specifies the HISTORY option for generated RUNSTATS utility statements. HISTORY specifies which statistics are to be recorded in the catalog history tables.

Values:

A

HISTORY ALL is added.

P

HISTORY ACCESSPATH is added.

S

HISTORY SPACE is added.

N

HISTORY NONE is added.

blank

The HISTORY option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

RUNSTATS (Db2 12 for z/OS)

UTIL_RUNSTATS_INVALIDATECACHE

Specifies the INVALIDATECACHE option for the generated RUNSTATS utility statements. INVALIDATECACHE specifies whether the dynamic statement cache is invalidated.

Values:

Y

INVALIDATECACHE YES is added to the utility statement.

N

INVALIDATECACHE NO is added to the utility statement.

blank

The INVALIDATECACHE option is not added to the utility statement. The Db2 default for this utility option is used.

Default:

blank

Related information:

RUNSTATS (Db2 12 for z/OS)

UTIL_RUNSTATS_PROFILE

Specifies whether the RUNSTATS utility is to use statistics profiles.

Values:

USE or U

USE PROFILE is added to the utility statement. USE PROFILE specifies that a stored statistics profile is to be used to gather statistics for a table.

DELETE or D

DELETE PROFILE is added to the utility statement. DELETE PROFILE specifies that the existing RUNSTATS profile for the table is to be deleted from the SYSIBM.SYSTABLES_PROFILES catalog table.

blank

Neither PROFILE clause is added to the utility statement.

Default:

blank

Related information:

RUNSTATS (Db2 12 for z/OS)

UTIL_RUNSTATS_REPORT

Specifies the REPORT option for generated RUNSTATS utility statements. REPORT specifies whether RUNSTATS is to generate a set of messages that report the collected statistics.

Values:

Y

REPORT YES is added.

N

REPORT NO is added.

blank

The REPORT option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

RUNSTATS (Db2 12 for z/OS)

UTIL_RUNSTATS_SHRLEVEL

Specifies the SHRLEVEL option for generated RUNSTATS utility statements. SHRLEVEL indicates whether the level of access that other programs have during utility processing.

Values:

C

SHRLEVEL CHANGE is added.

R

SHRLEVEL REFERENCE is added.

blank

The SHRLEVEL option is not added. The Db2 default for this utility option is used.

Default:

blank

Related information:

RUNSTATS (Db2 12 for z/OS)

UTIL_RUNSTATS_SHRLEVEL_CHANGE_REGISTER

Specifies whether, when SHRLEVEL CHANGE behavior is in effect, pages that are read by the RUNSTATS utility in a data sharing environment are registered with the coupling facility.

Values:

YES (or Y)

Pages are registered.

NO (or N)

Pages are not registered.

Default:

YES

Related information:

RUNSTATS (Db2 12 for z/OS)

UTIL_RUNSTATS_SORTDEVT

Specifies the device type that the sort program uses to dynamically allocate the sort work data sets that are required by the RUNSTATS utility.

Values:

device-type

The disk device type.

blank

SORTDEVT is not added to the utility statement.

Default:

blank

Related information:

RUNSTATS (Db2 12 for z/OS)

UTIL_RUNSTATS_SORTNUM

Specifies the number of required sort work data sets that the sort program is to allocate for the RUNSTATS utility.

Values:

integer

The number of temporary data sets. Valid values range from 2 to 255.

blank

SORTNUM is not added to the utility statement.

Default:

blank

Related information:

RUNSTATS (Db2 12 for z/OS)

UTIL_RUNSTATS_UPDATE

Specifies the UPDATE option for generated RUNSTATS utility statements. UPDATE indicates which collected statistics are to be inserted into the catalog tables.

Values:

A

UPDATE ALL is added.

P

UPDATE ACCESSPATH is added.

S

UPDATE SPACE is added.

N
UPDATE NONE is added.

blank
The UPDATE option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

RUNSTATS (Db2 12 for z/OS)

UTIL_UNLOAD_DBCS_CCSID
Specifies the DBCS value for the CCSID option for generated UNLOAD utility statements.

Values:

A valid CCSID for DBCS data
The CCSID option is added with the specified value as the third value of the CCSID parameters.

The CCSID is specified as follows: CCSID(**`UTIL_UNLOAD_SBCS_CCSID`**, **`UTIL_UNLOAD_MIXED_CCSID`**, **`UTIL_UNLOAD_DBCS_CCSID`**)

blank
The DBCS value is omitted from the CCSID option. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_ENCODINGSCHEME
Specifies which encoding scheme option (EBCDIC, ASCII, or UNICODE) to add to generated UNLOAD utility statements.

Values:

E
The EBCDIC option is added.

A
The ASCII option is added.

U
The UNICODE option is added.

blank
No encoding scheme option is added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_FLOAT
Specifies the FLOAT option for generated UNLOAD utility statements. FLOAT specifies the output format of the numeric floating-point data.

Values:

S
FLOAT S390 option is added.

I
FLOAT IEEE is added.

blank
The FLOAT option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_FORMAT_INTERNAL
Specifies the FORMAT INTERNAL option for generated UNLOAD utility statements. FORMAT INTERNAL specifies that the output record format is Db2 internal format.

Values:

YES
FORMAT INTERNAL is added.

blank
The FORMAT INTERNAL option is not added.

Default:
blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_IMPLICIT_TZ
Specifies the IMPLICIT_TZ option for generated UNLOAD utility statements. IMPLICIT_TZ specifies the implicit time zone to use when timestamp values are being unloaded from a TIMESTAMP column with no time zone

Values:

A valid IMPLICIT_TZ value for UNLOAD
The IMPLICIT_TZ option is added with the specified value.

blank
The IMPLICIT_TZ option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_MAXERR
Specifies the MAXERR option for generated UNLOAD utility statements. MAXERR specifies the maximum allowable number of records in error.

Values:

A valid MAXERR value for UNLOAD
The MAXERR option is added with the specified value.

blank
The MAXERR option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_MIXED_CCSID
Specifies the MIXED value for the CCSID option for generated UNLOAD utility statements.

Values:

A valid CCSID for mixed data

The CCSID option is added with the specified value as the second value of the CCSID parameters.

The CCSID is specified as follows: CCSID(**`UTIL_UNLOAD_SBCS_CCSID`**, **`UTIL_UNLOAD_MIXED_CCSID`**, **`UTIL_UNLOAD_DBCS_CCSID`**)

blank

The value for mixed data is omitted from the CCSID option. The Db2 default for this utility option is used.

Default:

blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_NOPAD

Specifies the NOPAD option for generated UNLOAD utility statements. NOPAD specifies that the variable-length columns are to be unloaded without additional padding.

Values:

Y

NOPAD is added.

N

The NOPAD option is not added.

Default:

N

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_NOSUBS

Specifies the NOSUBS option for generated UNLOAD utility statements. NOSUBS specifies that substitution characters are not to be used for any CCSID conversions that occur during unload processing.

Values:

Y

The NOSUBS option is added.

N

The NOSUBS option is not added.

Default:

N

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_PARALLEL

Specifies the PARALLEL option for generated UNLOAD utility statements. PARALLEL specifies the maximum number of subtasks that are to be used in parallel when unloading a partitioned table space.

Values:

YES

The PARALLEL option is added.

integer

The PARALLEL option is added to the utility statement with the specified value.

blank
The PARALLEL option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_SBCS_CCSID
Specifies the SBCS value for the CCSID option for generated UNLOAD utility statements.

Values:

A valid CCSID for SBCS data
The CCSID option is added with the specified value as the first value of the CCSID parameters.

The CCSID is specified as follows: CCSID(**UTIL_UNLOAD_SBCS_CCSID**, **UTIL_UNLOAD_MIXED_CCSID**, **UTIL_UNLOAD_DBCS_CCSID**)

blank
The SBCS value is omitted from the CCSID option. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_SHRLEVEL
Specifies the SHRLEVEL option for generated UNLOAD utility statements. SHRLEVEL specifies whether other processes can access or update the table space or partitions while the data is being unloaded.

Values:

1
SHRLEVEL CHANGE ISOLATION CS is added.

2
SHRLEVEL CHANGE ISOLATION UR is added.

3
SHRLEVEL REFERENCE is added.

blank
The SHRLEVEL option is not added. The Db2 default for this utility option is used.

Default:
blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_SHRLEVEL_CHANGE_REGISTER
Specifies whether, when ISOLATION UR and SHRLEVEL CHANGE behaviors are in effect, pages that are read by the UNLOAD utility in a data sharing environment are registered with the coupling facility.

Values:

YES (or Y)
Pages are registered.

NO (or N)
Pages are not registered.

Default:
blank

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_SKIP_LOCKED_DATA

Specifies the SKIP LOCKED DATA option for generated UNLOAD utility statements. SKIP LOCKED DATA specifies that the UNLOAD utility is to skip rows on which incompatible locks are held by other transactions.

Values:

YES

The SKIP LOCKED DATA option is added.

NO

The SKIP LOCKED DATA option is not added.

Default:

NO

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

UTIL_UNLOAD_SPANNED

Specifies the SPANNED value that is passed to the generated UNLOAD utility statements. SPANNED indicates whether records are to be unloaded into a VBS data set in spanned record format.

Values:

YES

SPANNED is added.

NO

SPANNED is not added.

Default:

No

Related information:

Syntax and options of the UNLOAD control statement (Db2 12 for z/OS)

Setting default CM batch parameter values by using profiles

You can establish and maintain your own default values for CM batch parameters. Individual invocations of CM batch can override these default values as needed.

About this task

When the Change Management (CM) batch interface is invoked, it reads parameters from the following two files in sequence: PROFPARM DD and then PARMS DD. PROFPARM DD contains the installation default values, and PARMS DD contains the individual invocation overrides. If you do not specify any defaults, the product default values are used.

Procedure

To set the default CM batch parameter values by using profiles:

- Define the PROFPARM DD in the JCL procedure.

  This parameter profile definition enables a JCL procedure parameter (for example, the SSID or the user-customized JCL procedure parameter) to dynamically determine which data set or data sets to associate with the parameter file in the JCL procedure (PROFPARM DD) .

- Define the PARMS DD when invoking the JCL procedure.

  Any parameter specified in the PARMS DD file overrides the parameter values in the PROFPARM DD file.

Examples:

The following JCL procedure defines the PROFPARM file and uses the Db2 SSID to determine which parameter profile to use:

```
//GOCCM   PROC SSID=,PLAN=,SPCUNIT=SYSDA
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(&SSID)
...
//GOCCM   PEND
```

When the CM batch interface is invoked, the SSID parameter value determines the member name in USERID.SSID.PARMS to use. For example, the following job invokes the JCL procedure:

```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=<DDL OR DELTA CHANGE FILE>
```

The value of the SSID parameter is DSNA. Therefore, the data set name for the `PROFPARM DD` in the JCL procedure resolves to the following definition:

```
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(DSNA)
```

Any parameter specified in the `PARMS DD` file overrides the parameter values in the `PROFPARM DD` file.

You can also define a user-customized JCL procedure parameter that determines an additional profile to use. For example:

```
//GOCCM   PROC SSID=,PLAN=,SPCUNIT=SYSDA,PROF=EMPTY
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(&SSID)
//          DD DISP=SHR,DSN=USERID.PROF.PARMS(&PROF)
...
//GOCCM   PEND
```

When the CM batch interface is invoked, the SSID parameter value determines the member name in USERID.SSID.PARMS to use. The PROF parameter value determines the member name in USERID.PROF.PARMS to use. For example, the following job invokes the JCL procedure:

```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB,PROF=LARGE
//GOCCM.PARMS  DD *
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=<DDL OR DELTA CHANGE FILE>
```

The SSID JCL parameter value is DSNA, and the user-defined JCL parameter PROF is LARGE. Therefore, the data set names for the `PROFPARM DD` in the JCL procedure resolve to the following definition:

```
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(DSNA)
//          DD DISP=SHR,DSN=USERID.PROF.PARMS(LARGE)
```

Any parameter specified in the `PARMS DD` file overrides what is specified in the `PROFPARM DD`.

Change Management reads the parameters in the following order:

1. USERID.SSID.PARMS(DSNA)

2. USERID.PROF.PARMS(LARGE)
3. The PARMS file

The value that CM uses for a parameter is the last one read.

Symbol variables in the CM batch interface

Symbol variables provide a method to define patterns for Change Management (CM) batch interface parameters related to items such as data set names, the new change owner, and the new change name.

The date-related and time-related symbol values are refreshed before saving or generating a base version. This enables a time-related variable, such as current timestamp (&CURTS.), to have different values when saving or generating multiple base versions in the same invocation of Change Management (CM) batch interface.

Related concepts

"Using user-defined symbol variables: Change Management (CM) batch interface" on page 795
You can define your own user-defined symbol variables and values to define patterns for Change Management (CM) batch interface parameters.

Related reference

"Product-defined symbol variables: Change Management (CM) batch interface" on page 792
The following table lists the product-defined symbol variables available in the Change Management (CM) batch interface. The value for each symbol variable is resolved at runtime.

"Symbol variables in the ADBTEMPL file: Db2 TEMPLATE support" on page 796
You can specify Db2 TEMPLATE statements in the ADBTEMPL file. References to specific symbol variables in the ADBTEMPL file are resolved by Db2 Admin Tool before the template statement is sent to Db2.

Product-defined symbol variables: Change Management (CM) batch interface

The following table lists the product-defined symbol variables available in the Change Management (CM) batch interface. The value for each symbol variable is resolved at runtime.

Symbol variables can be specified in all of the Change Management (CM) batch interface parameters:

Note: Time-related variables are resolved one time and remain the same value wherever they are used.

Table 36. Product-defined symbol variables for Change Management (CM) batch interface

Symbol variable	Description
&SSID. or &SS.	Subsystem ID
&CURSQLID.	CURRENT SQLID
&CURTS.	CURRENT TIMESTAMP
&DATE. or &DT.	YYYYDDD
&JDAY. or &JD.	DDD portion of &DATE.
&JOBNAME. or &JO.	
&USERID. or &US.	The user ID of the person. who is running the job
&YEAR. or &YE.	YYYY
&MONTH. or &MO.	MM
&DAY. or &DA.	DD
&TIME. or &TI.	HHMMSS
&HOUR. or &HO.	HH portion of &time.
&MINUTE. or &MI.	MM portion of &time.

Table 36. Product-defined symbol variables for Change Management (CM) batch interface (continued)

Symbol variable	Description
&SECOND. or &SC.	SS portion of &time.
&CHGTAG.	
&CHGTAG. (continued)	When the chgtag_type is NAME, the &CHGTAG. symbol variable resolves to values based on the user specified change name: • When processing the original change, the change name for data set names. The member name is the original change name for both the WSL PDS member and the run JCL PDS member. • When processing the recover change, the change name of the original change with '.R' appended for data set names. The member name is the original change name for both the recover WSL PDS member and the recover JCL PDS member. When the chgtag_type is OWNER, the &CHGTAG. symbol variable resolves to values based on the user specified change owner: • When processing the original change, the change owner for data set names. The member name is the original change owner for both the WSL PDS member and the run JCL PDS member. • When processing the recover change, the change owner of the original change with '.R' appended for data set names. The member name is the original change owner for both the recover WSL PDS member and the recover JCL PDS member.

Change tag (&CHGTAG.) Usage: chgtag_type = 'NAME'

Using the change name instead of the change ID can be helpful when you want to automate portions of your change management process, or when you want to more easily locate data sets associated with a change. However, be aware of the restrictions. For example, the change name of the original change must be less than or equal to 8 characters, and conform to z/OS data set naming rules.

⚠ **Attention:** It is critical that you ensure that each registered change on a Db2 subsystem is unique by its change name only. The change name is used in data set names and common PDS member names. There might be two changes with the same change name but with different change owners on the same Db2 subsystem. Failing to ensure unique change names for all registered changes can result in change artifacts in a data set being overwritten by another change with the same change name. Db2 Admin Tool enforces the uniqueness by change owner and change name, but not by the change name alone. If all users of Db2 Admin Tool Change Management use the same change owner, then Db2 Admin Tool ensures that the change name is unique for all registered changes on a Db2 subsystem.

⚠ **Attention:** If the same PDS is used to store change artifacts for multiple Db2 subsystems (for example, the run JCL PDS), you must ensure that the change name is unique across all Db2 subsystems that share the PDS. Take extra care to ensure that you have unique change names across multiple Db2 subsystems.

When a recover change is requested, the PDS member name is the same for both the original and recover change. The following data sets for a recover change must be different from the original change data sets after symbols are resolved:

- pds_for_recover_jcl must be different from pds_for_run_jcl
- pds_for_recover_wsl must be different from pds_for_wsl

There is a change in behavior to the Db2 Admin Tool skeleton template data set name customization, specifically to skeletons ADB2UCUS and ADB2UCUU. When you are generating the recover change, the &LEVEL symbol in the ADB2UCUS and ADB2UCUU skeletons is 2 characters more than the value of the change name of the original change. Because the maximum length of the change name is 8, &LEVEL must be a maximum length of 10, instead of the normal maximum length of 8. For example, if the change_name is ABCDEFGH (character length of 8), then when you are generating the recover change, the &LEVEL resolves to ABCDEFGH.R (a character length of 10).

Change tag (&CHGTAG.) Usage: chgtag_type = 'OWNER'

Use of the change owner instead of the change name provides more flexibility for user customized environments in which the change owner and change name have different meanings. Restrictions and considerations when you specify the chgtag_type as OWNER is similar to chgtag_type as NAME. The change owner of the original change must be less than or equal to 8 characters, and conform to z/OS data set naming rules.

⚠ **Attention:** It is critical that you ensure that each registered change on a Db2 subsystem is unique by its change owner only. The change owner is used in data set names and common PDS member names. There might be two changes with the same change owner but with different change names on the same Db2 subsystem. Failing to ensure unique change owners for all registered changes can result in change artifacts in a data set being overwritten by another change with the same change owner. Db2 Admin Tool enforces the uniqueness by change owner and change name, but not by the change owner alone. If all users of Db2 Admin Tool Change Management use the same change name, then Db2 Admin Tool ensures that the change owner is unique for all registered changes on a Db2 subsystem.

⚠ **Attention:** If the same PDS is used to store change artifacts for multiple Db2 subsystems (for example, the run JCL PDS), you must ensure that the change owner is unique across all Db2 subsystems that share the PDS. Take extra care to ensure that you have unique change owners across multiple Db2 subsystems.

When a recover change is requested, the PDS member name is the same for both the original and recover change. The following data sets for a recover change must be different from the original change data sets after symbols are resolved:

- pds_for_recover_jcl must be different from pds_for_run_jcl
- pds_for_recover_wsl must be different from pds_for_wsl

There is a change in behavior to the Db2 Admin Tool skeleton template data set name customization, specifically to skeletons ADB2UCUS and ADB2UCUU. When you are generating the recover change, the &LEVEL symbol in the ADB2UCUS and ADB2UCUU skeletons is 2 characters more than the value of the change owner of the original change. Because the maximum length of the change name is 8, &LEVEL must be a maximum length of 10, instead of the normal maximum length of 8. For example, if the change_owner is ABCDEFGH (character length of 8), then when you are generating the recover change, the &LEVEL resolves to ABCDEFGH.R (a character length of 10).

&CHGTAG. examples

chgtag_type = 'ID'

When an original change consists of a Db2 Admin Tool generated change ID of 45, and the user specified change name is ABCDEFGH, &CHGTAG. resolves to C0000045, when files are generated for the original change. The run JCL PDS member name is E0000045. When files are generated for the recover change, &CHGTAG. resolves to R0000045. The recover JCL PDS member name is R0000045. Assuming default values are used for the data set names, the following is a subset of the data set names that are used for the original change:

```
USERID.SSID.C0000045.CHG
USERID.SSID.C0000045.IFF
USERID.SSID.RUN.WSL(C0000045)
USERID.SSID.RUN.JCL(E0000045)
USERID.SSID.C0000045.IN
```

The following is a subset of the data set names that are used for the recover change:

```
USERID.SSID.R0000045.CHG
USERID.SSID.R0000045.IFF
USERID.SSID.RECOVER.WSL(R0000045)
USERID.SSID.RECOVER.JCL(R0000045)
USERID.SSID.R0000045.IN
```

chgtag_type = 'NAME'

When an original change consists of a Db2 Admin Tool generated change ID of 45, and the user specified change name is ABCDEFGH, the original change name of ABCDEFGH is used as the PDS member name for the JCL and WSL PDS members. When generating data set names for the original change, &CHGTAG. resolves to the original change name. When generating data set names for the recover change, &CHGTAG. resolves to the original change name with '.R' appended. Assuming default values are used for the data set names, the following is a subset of the data set names that are used for the original change:

```
USERID.SSID.ABCDEFGH.CHG
USERID.SSID.ABCDEFGH.IFF
USERID.SSID.RUN.WSL(ABCDEFGH)
USERID.SSID.RUN.JCL(ABCDEFGH)
USERID.SSID.ABCDEFGH.IN
```

The following is a subset of the data set names that are used for the recover change:

```
USERID.SSID.ABCDEFGH.R.CHG
USERID.SSID.ABCDEFGH.R.IFF
USERID.SSID.RECOVER.WSL(ABCDEFGH)
USERID.SSID.RECOVER.JCL(ABCDEFGH)
USERID.SSID.ABCDEFGH.R.IN
```

Using user-defined symbol variables: Change Management (CM) batch interface

You can define your own user-defined symbol variables and values to define patterns for Change Management (CM) batch interface parameters.

About this task

You can use user-defined symbol variables in any parameter that a product-defined symbol variable can be specified. For a list of parameters that support product-defined symbol variables, see "Product-defined symbol variables: Change Management (CM) batch interface" on page 792.

Procedure

1. To learn how to use user-defined symbol variables, refer to the following examples.

Example 1: Defining the symbol &TASKNUM.

Suppose you define a symbol &TASKNUM. with a value of A123. &TASKNUM. could be referenced in the parameters like the following:

- prefix_for_data_sets: &USERID..&TASKNUM.
- pds_for_wsl: &SSID..ANALYZE.WSL
- pds_for_jcl: &SSID..ANALYZE.JCL
- new_change_owner: &CURSQLID.

- new_change_name: &TASKNUM.-&CURTS.

```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
New_change_name = ‘&TASKNUM.-&CURTS.’;
Symbol_name=’&TASKNUM.’, symbol_value=’A123’;
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=<DDL OR DELTA CHANGE FILE>
```

Example 2: Using PROFPARM, PARMS, and user-defined symbols

In file USERID.SSID.PARMS(DSNA), the following parameter is specified using a user-defined symbol &TASKNUM.:

```
New_change_name = ‘&TASKNUM.-&CURTS.’;
```

In the JCL procedure for Change Management (CM) batch interface (GOCCM), the PROFPARM file is defined like the following:

```
//GOCCM   PROC SSID=,PLAN=,SPCUNIT=SYSDA
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(&SSID)
…
//GOCCM   PEND
```

In the call to the Change Management (CM) batch interface, the PARMS file is defined and the &TASKNUM. symbol is defined as the work order # A123.

```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
Symbol_name=’&TASKNUM.’, symbol_value=’A123’;
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=<DDL OR DELTA CHANGE FILE>
```

When the Change Management (CM) batch interface is invoked the PROFPARM file gets resolved to:

```
//PROFPARM  DD DISP=SHR,DSN=USERID.SSID.PARMS(DSNA)
```

When a new change is created, the change name is something like A123-2011-11-15-22.52.05.42333.

Symbol variables in the ADBTEMPL file: Db2 TEMPLATE support

You can specify Db2 TEMPLATE statements in the ADBTEMPL file. References to specific symbol variables in the ADBTEMPL file are resolved by Db2 Admin Tool before the template statement is sent to Db2.

References to the following symbol variables are resolved by Db2 Admin Tool:

- User-defined symbol variables that are defined in the CM Batch parameter list.
- The &CHGTAG. product-defined symbol variable.
- The following product-defined symbol variables that are only resolved when referenced in the ADBTEMPL file:

Table 37. Symbol variables that are resolved only when referenced in the ADBTEMPL file

Symbol variable	Description
&PREFIX.	The value of. the prefix_for_data_sets parameter
&TNAME.	

Substring notation is not supported for Db2 Admin Tool and user-defined symbol variables.

Importing changes by using the Change Management (CM) batch interface

To import changes to multiple target locations, you can run one Change Management batch job that imports the changes to all of the subsystems at the same time.

If your subsystems have DRDA connectivity, the changes are automatically imported across subsystems. If your subsystems do not have DRDA connectivity and are using FILE as the communication method, you must manually import the changes, one subsystem at a time.

Before you begin

To import multi-target changes, you must have either a DDL file that contains all of the object definitions or a changes file that contains the delta change statements that you want to apply to your targets. The following example shows a sample DDL file:

```
BROWSE SYSADM.DDL(XDB) - 01.00           Line 0000000000 Col 001 080
Command ===>                                        Scroll ===> CSR
********************************* Top of Data **********************************
  CREATE DATABASE XDB
    BUFFERPOOL BP0
    INDEXBP    BP1
    CCSID      EBCDIC
    STOGROUP   SYSDEFLT;
******************************** Bottom of Data ********************************
```

You must also have masks defined for your targets, as described in “Creating masks in the Change Management repository” on page 305.

About this task

For the examples in this procedure, subsystem DSNA is the central system, and the target locations are DSNA and DSNB. The masks SYSADM.MASK A1and SYSADM.MASK A2 are defined on subsystem DSNA. The mask SYSADM.MASK B is defined on subsystem DSNB. SYSADM.MASK A1 is defined as:

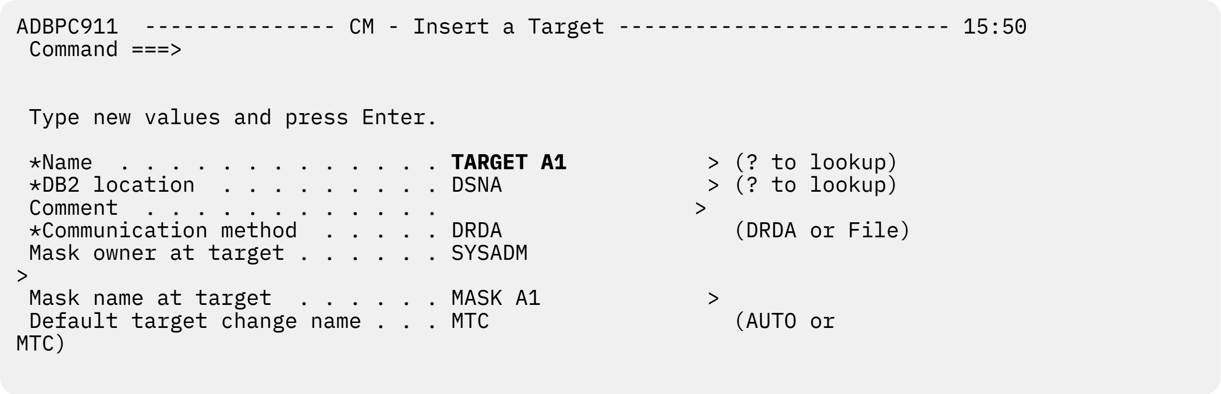
```
DBNAME :XDB,A1DB
```

Procedure

To import changes by using the Change Management (CM) batch interface:

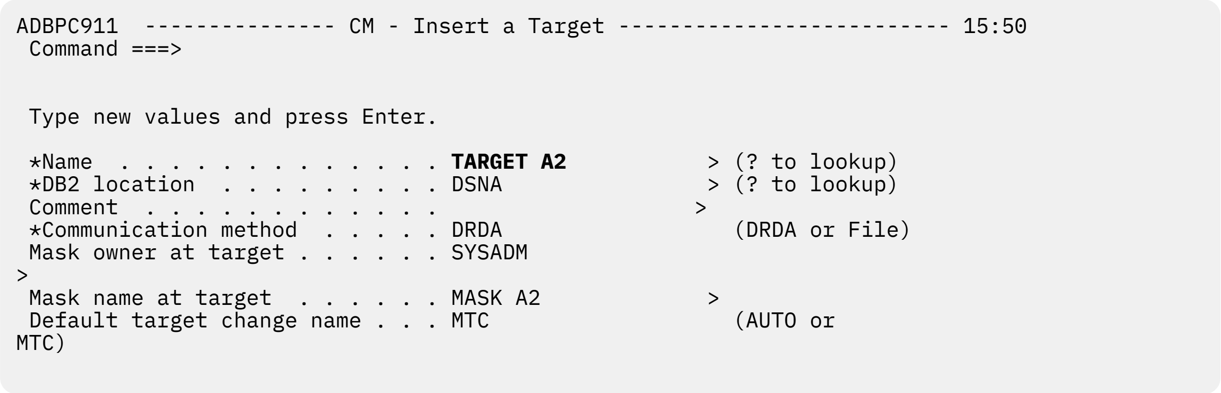
- If your subsystems have DRDA connectivity, complete the following steps:
  a) Ensure that you are on the central system from which you want to initiate the multi-target change. The central system tracks the changes that you import to the target systems.
  b) On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
  c) On the **Change Management (CM) (ADB2C)** panel, specify option 9, and press Enter.
  d) On the **CM - Manage Targets (ADBPC9)** panel, specify option 3, and press Enter.
  e) On the **CM - Insert a Target (ADBPC911)** panel, define a target or targets.

The target locations are the Db2 locations of the remote server. You must define each target to which you want to import changes. In this example, TARGET A1, TARGET A2, and TARGET B are defined, as shown in the following figures:



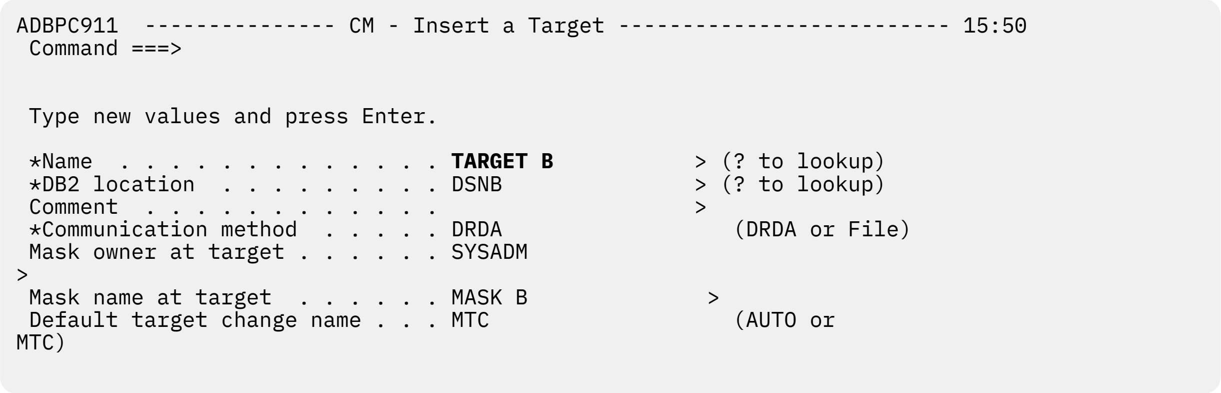
```
ADBPC911  --------------- CM - Insert a Target -------------------------- 15:50
 Command ===>


 Type new values and press Enter.

 *Name  . . . . . . . . . . . . . TARGET A1           > (? to lookup)
 *DB2 location  . . . . . . . . . DSNA                > (? to lookup)
 Comment  . . . . . . . . . . . .                    >
 *Communication method  . . . . . DRDA                  (DRDA or File)
 Mask owner at target . . . . . . SYSADM
>
 Mask name at target  . . . . . . MASK A1             >
 Default target change name . . . MTC                   (AUTO or
MTC)
```

Figure 367. Defining Target A1



```
ADBPC911  --------------- CM - Insert a Target -------------------------- 15:50
 Command ===>


 Type new values and press Enter.

 *Name  . . . . . . . . . . . . . TARGET A2           > (? to lookup)
 *DB2 location  . . . . . . . . . DSNA                > (? to lookup)
 Comment  . . . . . . . . . . . .                    >
 *Communication method  . . . . . DRDA                  (DRDA or File)
 Mask owner at target . . . . . . SYSADM
>
 Mask name at target  . . . . . . MASK A2             >
 Default target change name . . . MTC                   (AUTO or
MTC)
```

Figure 368. Defining Target A2



```
ADBPC911  --------------- CM - Insert a Target -------------------------- 15:50
 Command ===>


 Type new values and press Enter.

 *Name  . . . . . . . . . . . . . TARGET B           > (? to lookup)
 *DB2 location  . . . . . . . . . DSNB               > (? to lookup)
 Comment  . . . . . . . . . . . .                    >
 *Communication method  . . . . . DRDA                  (DRDA or File)
 Mask owner at target . . . . . . SYSADM
>
 Mask name at target  . . . . . . MASK B              >
 Default target change name . . . MTC                   (AUTO or
MTC)
```

Figure 369. Defining Target B

f) Create a Change Management batch JCL job. In the Change Management batch file, specify values for the following parameters:

- ACTION_IMPORT_CHANGE = 'Y'. (You must set this parameter to Y to import the changes.)
- CHANGE_OWNER
- CHANGE_NAME
- TARGET_PROFILE_NAME
- TARGET_MASK_OWNER
- TARGET_MASK_NAME

- TARGET_CHANGE_OWNER
- TARGET_CHANGE_NAME

The following example shows a sample batch file:

```
<JOB CARDS>
//*
//ADBLIBS JCLLIB ORDER=<CM batch PROCLIB>
//*
//***********************************************************GOCCM*******
//* STEP TO RUN A MULTI-TARGET CHANGE
//***********************************************************************
//ANLYZ EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
ACTION_IMPORT_CHANGE='Y'
ACTION_ANALYZE_CHANGE='N'
IMPORT_PENDING_CHANGE_ACTION='S'
CHANGE_OWNER='SYSADM'
CHANGE_NAME='CENTRAL MTC CHANGE - FOR TARGETS A1, A2 & B'
;
TARGET_PROFILE_NAME='TARGET A1'
,TARGET_MASK_NAME='MASK A1'
,TARGET_CHANGE_OWNER='SYSADM'
,TARGET_CHANGE_NAME='CHANGE1 FOR TARGET A1';
TARGET_PROFILE_NAME='TARGET A2'
,TARGET_MASK_OWNER='SYSADM'
,TARGET_MASK_NAME='MASK A2'
,TARGET_CHANGE_OWNER='SYSADM'
,TARGET_MASK_NAME='MASK A2'
,TARGET_CHANGE_OWNER='SYSADM'
,TARGET_CHANGE_NAME='CHANGE2 FOR TARGET A2';
TARGET_PROFILE_NAME='TARGET B'
TARGET_MASK_OWNER='SYSADM'
,TARGET_CHANGE_NAME='CHANGE2 FOR TARGET A2';
TARGET_PROFILE_NAME='TARGET B'
,TARGET_MASK_OWNER='SYSADM'
,TARGET_MASK_NAME='MASK B'
,TARGET_CHANGE_OWNER='SYSADM'
,TARGET_CHANGE_NAME='CHANGE3 FOR TARGET
B';
/*
//IMCHG001 DD DISP=SHR,
//          DSN=SYSADM.DDL(XDB)
//MTC      DD DISP=SHR,
//          DSN=SYSADM.MTC
//        ENDIF
```

g) Run the batch job.

After the batch job runs successfully, a multi-target import report is generated, as shown in the following example:

```
********************************* TOP OF DATA **********************************
-------------------------------------------------------------------------------
ADB2CID - Import changes - 2015-11-11 15:07
-------------------------------------------------------------------------------


ADB2CID - Multi-Target Change Summary

Multi-target change id:           7913

Target    Owner    Name               Status
--------- -------- ------------------ ------------------------------------------
TARGET A1 SYSADM   CHANGE1 FOR TARGET ADB9400I:The change was registered successfully,
Changeid: 7914
TARGET A2 SYSADM   CHANGE2 FOR TARGET ADB9400I:The change was registered successfully,
Changeid: 7915
TARGET B  SYSADM   CHANGE3 FOR TARGET ADB9400I:The change was registered successfully,
Changeid: 4415

ADB2CID - Multi-Target Change End of Summary

ADB0004I ADBCCM - Ended normally
******************************** BOTTOM OF DATA ********************************
```

h) Optional: Verify that your imports were successful by completing the following steps:

a. On the **Change Management (CM) (ADB2C)** panel, select option 1.

b. On the **Manage Changes (ADB2C1)** panel, select option 1.

c. On the **CM - Changes (ADB2C11)** panel, issue the AT line command for the multi-target change to see the status of the target changes, as shown in the following figure:

```
ADB2C11 n ------------------ CM - Changes -------------------- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Commands: COMMENT  EXPORT  REFRESH
Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks  S - Show WSL  B - Checkpoint
 ? - Show all line commands

Sel       ID Owner    Name                         Type     Status   I Comment
           * *        *                            *        *        * *
--- -------- -------- -------------------------> -------- -------- - ---------
        7915 SYSADM   CHANGE2 FOR TARGET A2       CHANGE   DEFINED
        7914 SYSADM   CHANGE1 FOR TARGET A1       CHANGE   DEFINED
AT      7913 SYSADM   CENTRAL MTC CHANGE - FOR TA MULTI-TC DEFINED
******************************* END OF DB2 DATA *******************************
```

Figure 370. ***CM - Changes (ADB2C11)*** *panel*

The **CM - Associate Targets (ADBPCMT)** panel shows the targets to which your change was successfully imported:

```
ADBPCMT n --------------- CM - Associate Targets ------------- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Details for multi-target change: SYSADM.CENTRAL MTC   > DB2 System: DSNA
                                                        DB2 SQL ID: SYSADM
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
    *        *             *         *                      *
--- -------> ------------> --------- ---------------------- -------------------
    TARGET A DSNA          SYSADM    CHANGE1 FOR TARGET A1  DEFINED
    TARGET A DSNA          SYSADM    CHANGE2 FOR TARGET A2  DEFINED
    TARGET B DSNB          SYASDM    CHANGE3 FOR TARGET B   DEFINED
******************************* END OF DB2 DATA *******************************
```

Figure 371. ***CM - Associate Targets (ADBPCMT)*** *panel*

- If your subsystems do not have DRDA connectivity, run a Change Management batch JCL job for each subsystem.

  In general, if the JCL procedure was set up so that the SSID parameter determines the Db2 libraries for the subsystem, the same CM batch JCL procedure can be used to import the change into the different subsystems.

  The following examples show how to import a change into multiple Db2 subsystems, one subsystem at a time.

  Example 1: Import to DSNA

```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
```

```
  Symbol_name=’&TASKNUM.’, symbol_value=’A123’;
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=DELTA.CHANGE(A123)
```

Example 2: Import to DSNB

```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNB,PLAN=ADB
//GOCCM.PARMS  DD *
  Symbol_name=’&TASKNUM.’, symbol_value=’A123’;
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=DELTA.CHANGE(A123)
```

Example 3: Import to DSNC

```
//DEMO     JOB (&SYSUID,ICE,ICE,ICE),'DEMO',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=JCL.PROCLIB
//*
//GOCCM    EXEC GOCCM,SSID=DSNC,PLAN=ADB
//GOCCM.PARMS  DD *
  Symbol_name=’&TASKNUM.’, symbol_value=’A123’;
/*
//GOCCM.IMCHG001 DD DISP=SHR,DSN=DELTA.CHANGE(A123)
```

Analyzing a multi-target change

You can analyze all multi-target changes that are registered on a central system by using the Change Management (CM) batch interface. Analyzing changes in batch is a more efficient alternative to analyzing changes one at a time through the CM panels.

About this task

Use the following procedure to manually define a batch job for analyzing a multi-target change. As an alternative to this manual procedure, you can use panels to create a CM batch analyze job; for instructions, see “Analyzing a multi-target change in batch by using panels” on page 803.

Procedure

To analyze a multi-target change:

1. In the CM batch file, specify ACTION_ANALYZE_CHANGE = 'Y', as shown in the following example:

```
<JOB CARDS>
//*
//ADBLIBS JCLLIB ORDER=<CM batch PROCLIB>
//*
//**********************************************************GOCCM*******
//* STEP ANALYZE A CHANGE
//**********************************************************************
//ANLYZ EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
 ACTION_IMPORT_CHANGE='N'
 ACTION_ANALYZE_CHANGE='Y'
 CHANGE_OWNER='SYSADM'
 CHANGE_NAME='CENTRAL MTC CHANGE - FOR TARGETS A & B'
/*
```

You can specify additional CM batch parameters as needed.

Tip: If the multi-target change exists on another subsystem, use the MTC_CENTRAL_LOCATION parameter to specify the location of the change and use the CHANGE_NAME parameter to specify the name of the change at the central location. If you do not specify a central location, you must run an analyze job with a unique change name for each target at each remote location.

Restriction: Currently, the GENERATE_RECOVER_CHANGE parameter is not supported when analyzing a multi-target change

2. Run the batch job.

   After the batch job runs successfully, a multi-target analyze report is generated, as shown in the following example:

```
--------------------------------------------------------------------------------
ADBCCM - Multi-target Change Summary Report (Analyze Process)
--------------------------------------------------------------------------------

Multi-target change id: 7913

Successful entries:

Location: DSNA

Target     ID     Owner    Name                    Status   Remarks
---------- ------ -------- ----------------------- -------- ---------------
TARGET A1  7914   SYSADM   CHANGE1 FOR TARGET A1   ANALYZED Analyze Successful
TARGET A2  7915   SYSADM   CHANGE2 FOR TARGET A2   ANALYZED Analyze Successful


Skipped or Failed or Running entries:

Target     Loc    ID     Owner    Name                  Status  Remarks
---------- ------ ------ -------- --------------------- ------- --------------
TARGET B   DSNB   N/A    SYSADM   CHANGE3 FOR TARGET B  DEFINED Skipped.
                                                                Target not on
                                                                local subsystem.


-------------------------------------------------------------------------------
ADBCCM - End of Multi-target Change Summary Report (Analyze Process)
-------------------------------------------------------------------------------

ADB0004I ADBCCM - Ended normally
******************************* BOTTOM OF DATA ********************************
```

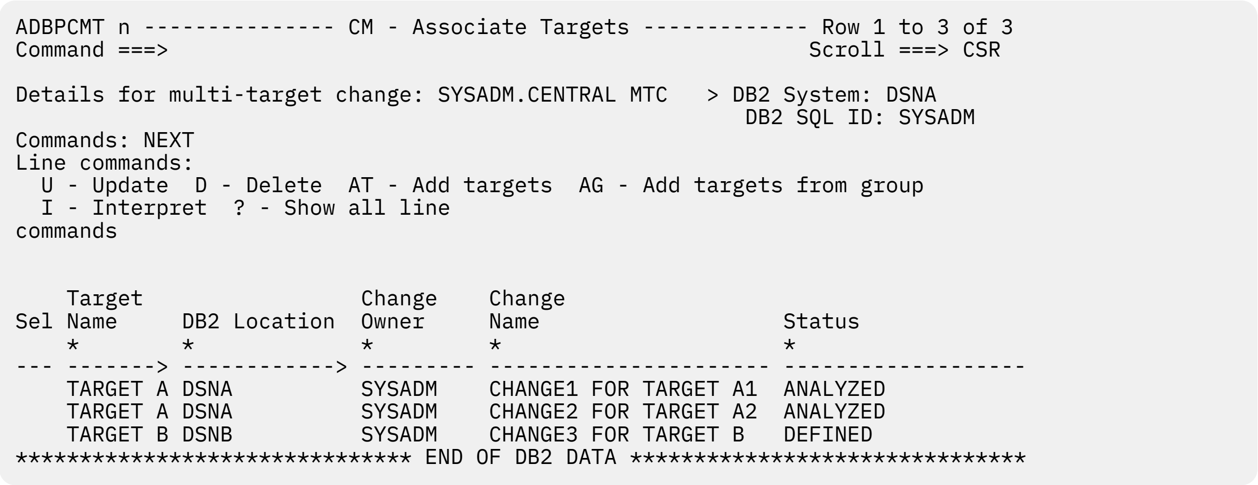
   In this example, TARGET B was skipped because it is not on the local subsystem, DSNA. You must use subsystem DSNB to analyze the change for TARGET B.

What to do next

After analyzing your changes, you might want to verify that your analyze was successful by completing the following steps:

1. On the **Change Management (CM) (ADB2C)** panel, specify option 1 (Manage changes), and press Enter.
2. On the **Manage Changes (ADB2C1)** panel, specify option 1 (Display changes), and press Enter.
3. On the **Changes (ADB2C11)** panel, issue the AT line command next to the multi-target change, and press Enter.

   The **CM - Associate Targets (ADBPCMT)** panel displays the status of all your target changes, as shown in the following example:



```
ADBPCMT n --------------- CM - Associate Targets ------------- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Details for multi-target change: SYSADM.CENTRAL MTC   > DB2 System: DSNA
                                                         DB2 SQL ID: SYSADM
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
    *        *             *         *                      *
--- -------> ------------> --------- ---------------------- -------------------
    TARGET A DSNA          SYSADM    CHANGE1 FOR TARGET A1  ANALYZED
    TARGET A DSNA          SYSADM    CHANGE2 FOR TARGET A2  ANALYZED
    TARGET B DSNB          SYSADM    CHANGE3 FOR TARGET B   DEFINED
******************************* END OF DB2 DATA *******************************
```

Figure 372. ***CM - Associate Targets (ADBPCMT)*** *panel*

Analyzing a multi-target change in batch by using panels

You can use panels to create a Change Management (CM) batch job to analyze multi-target changes. Using a CM batch job can be more efficient than analyzing changes one at a time with the CM panels.

Procedure

To use panels to create a multi-target analyze batch job:

1. On the **Change Management (CM) (ADB2C)** panel, specify option 1 (Manage changes), and press Enter.
2. On the **Manage Changes (ADB2C1)** panel, specify option 1 (Display changes), and press Enter.
3. On the **Changes (ADB2C11)** panel, issue one of the following line commands next to the change that you want to analyze:

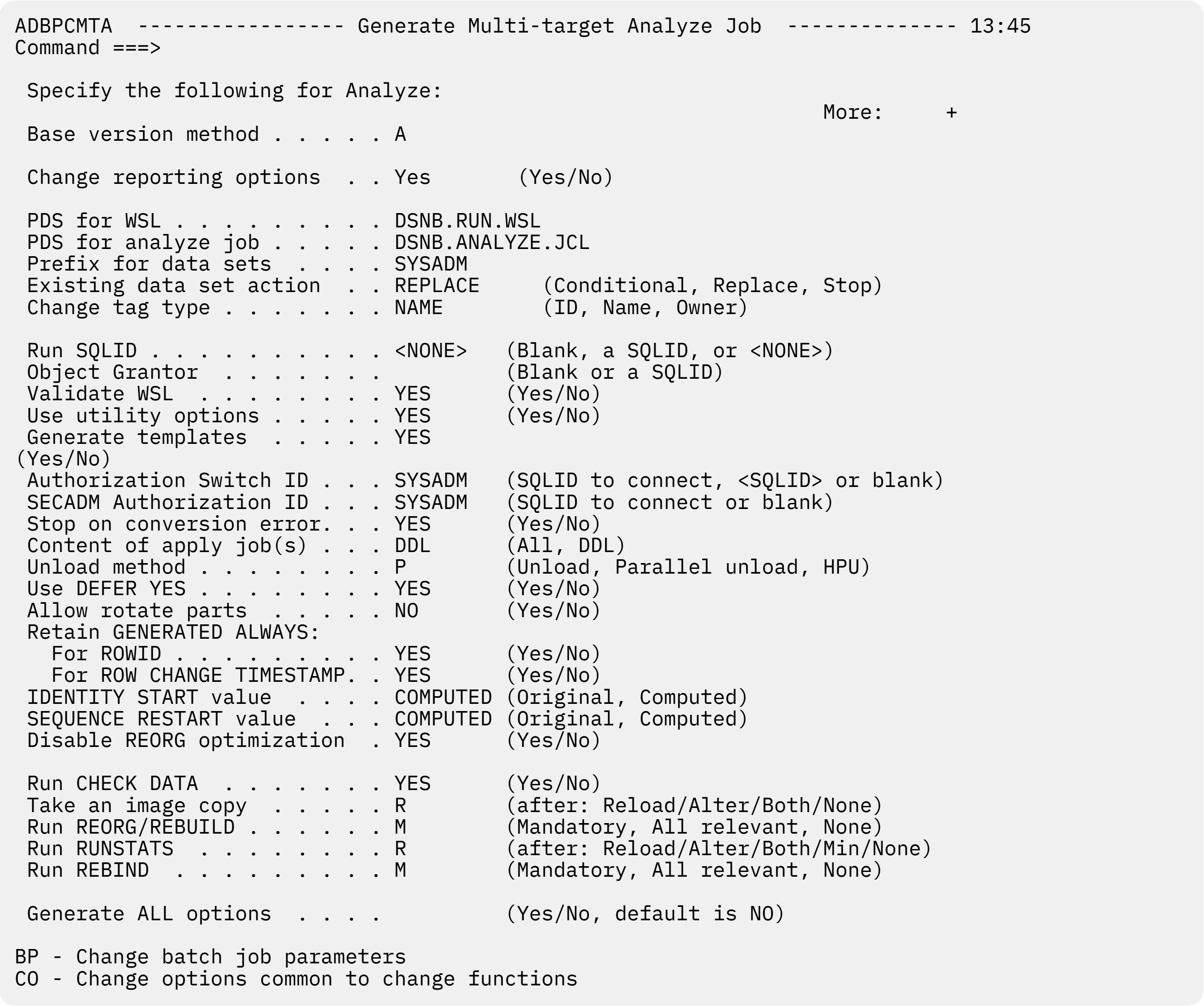
   AN

   Creates a simple CM batch job without any panel options

   ANO

   Allows you to specify analyze options to include in the generated CM job. ANO can be specified only if the change type is MULTI-TC

   .
4. On the **Generate Multi-target Analyze Job (ADBPCMTA)** panel, complete the fields, and press Enter:



```
ADBPCMTA  ---------------- Generate Multi-target Analyze Job  -------------- 13:45
Command ===>

 Specify the following for Analyze:
                                                                  More:     +
 Base version method . . . . . A

 Change reporting options  . . Yes       (Yes/No)

 PDS for WSL . . . . . . . . . DSNB.RUN.WSL
 PDS for analyze job . . . . . DSNB.ANALYZE.JCL
 Prefix for data sets  . . . . SYSADM
 Existing data set action  . . REPLACE     (Conditional, Replace, Stop)
 Change tag type . . . . . . . NAME        (ID, Name, Owner)

 Run SQLID . . . . . . . . . . <NONE>   (Blank, a SQLID, or <NONE>)
 Object Grantor  . . . . . . .          (Blank or a SQLID)
 Validate WSL  . . . . . . . . YES      (Yes/No)
 Use utility options . . . . . YES      (Yes/No)
 Generate templates  . . . . . YES
(Yes/No)
 Authorization Switch ID . . . SYSADM   (SQLID to connect, <SQLID> or blank)
 SECADM Authorization ID . . . SYSADM   (SQLID to connect or blank)
 Stop on conversion error. . . YES      (Yes/No)
 Content of apply job(s) . . . DDL      (All, DDL)
 Unload method . . . . . . . . P        (Unload, Parallel unload, HPU)
 Use DEFER YES . . . . . . . . YES      (Yes/No)
 Allow rotate parts  . . . . . NO       (Yes/No)
 Retain GENERATED ALWAYS:
   For ROWID . . . . . . . . . YES      (Yes/No)
   For ROW CHANGE TIMESTAMP. . YES      (Yes/No)
 IDENTITY START value  . . . . COMPUTED (Original, Computed)
 SEQUENCE RESTART value  . . . COMPUTED (Original, Computed)
 Disable REORG optimization  . YES      (Yes/No)

 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)

 Generate ALL options  . . . .          (Yes/No, default is NO)

BP - Change batch job parameters
CO - Change options common to change functions
```

*Figure 373. **Generate Multi-target Analyze Job (ADBPCMTA)** panel*

The first time this panel is used in a session, values are pre-filled with the CM batch default parameter values. For subsequent uses of this panel in a session, the previously entered values are reused. To reset the panel to use the default values, issue the DEFAULTS command. This command overrides any previously entered values.

Most fields on the panel correspond to CM batch parameters; see "CM batch parameter definitions" on page 664.

Batch parameters are generated in the job only if the corresponding value entered on the panel is different from the CM batch default value. If you want to generate CM batch options for all values that are specified on the panel, regardless of whether the values differ from the default values, set the **GENERATE ALL options** field to Y.

Note: If a target or targets exist on other subsystems, the generated jobs include additional steps for remote subsystems with the MTC_CENTRAL_LOCATION parameter and the appropriate location name of the change. In this case, the PDS_FOR_WSL parameter is generated with the default value to avoid overwriting members in the same data set by different steps.

Depending on the values that you specify, you might be prompted for additional information before the analyze job is generated and the ISPF Edit session is displayed.

Results

A CM batch job is generated in the data set that is specified in the **PDS for analyze job** field, as shown in the following example:

```
****** ***************************** Top of Data ******************************
000001 //RHP1 JOB ,
000002 //*       RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
000003 //     REGION=0M,NOTIFY=&SYSUID,
000004 //     MSGCLASS=H,TIME=(,30)
000005 //*
000006 /*JOBPARM S=SY4A
000007 //*
000008 //ADBLIBS JCLLIB ORDER=ADB.DEVCUST.PROCLIB
000009 //*
000010 //******************************************************GOCCM***********
000011 //* STEP REGISTER A CHANGE
000012 //**********************************************************************
000013 //GOCCM EXEC GOCCM,SSID=DSNA,PLAN=ADBDEV,GRP=VB2APAR
000014 //GOCCM.PARMS  DD *
000015  ACTION_ANALYZE_CHANGE='Y'
000016  CHANGE_OWNER='SYSADM'
000017  CHANGE_NAME='TESTRHP_CHG03'
000018
000019  PREFIX_FOR_DATA_SETS = ’SYSADM’
000020  PDS_FOR_WSL = ’DSNB.RUN.WSL’
000021  EXISTING_DATA_SET_ACTION=’REPLACE’
000022  CHGTAG_TYPE = 'NAME'
000023
000023  RUN_SQLID = '<NONE>'
000024  VALIDATE_WSL = 'Y'
000025  USE_UTILITY_OPTIONS='Y'
000025  GENERATE_TEMPLATES = 'Y'
000026  AUTH_SWITCH_USERID = 'SYSADM'
000027  AUTH_SWITCH_SECADM = 'SYSADM'
000028  STOP_ON_CONVERSION_ERROR = 'Y'
000029  CONTENT_OF_APPLY_JOBS = 'D'
000030  UNLOAD_METHOD = 'P'
000031  USE_DEFER_YES = 'Y'
000032  ALLOW_ROTATE_PARTS = 'N'
000033  RETAIN_GENERATED_ALWAYS_FOR_ROWID = 'Y'
000034  RETAIN_GENERATED_ALWAYS_FOR_ROW_CHANGE_TS = 'Y'
000035  IDENTITY_START_VALUE = 'C'
000036  SEQUENCE_RESTART_VALUE = 'COMPUTED'
000037  DISABLE_OPTIMIZE_REORG = 'Y'
000038
000039  RUN_CHECK_DATA = 'Y'
000040  TAKE_AN_IMAGE_COPY = 'R'
000041  RUN_REORG_REBUILD = 'M'
000042  RUN_RUNSTATS = 'R'
000043  RUN_REBIND = 'M'
000044
000045  REPORT_ONLY_CHANGED_OBJECTS = 'Y'
000046  REPORT_USER_SPECIFIED_IGNORE_FIELDS = 'Y'
000047  REPORT_SYSTEM_GENERATED_IGNORE_FIELDS = 'Y'
000048  REPORT_TRANSLATION_MASKS = 'Y'
000049  REPORT_SUMMARY = 'Y'
000050  REPORT_OBJECT_COUNT = 'Y'
000051  REPORT_EXPECTED_CONVERSION_PROBLEMS = 'Y'
000052
000053  RELOAD_ACCELERATED_TABLES = 'N'
000054  ;
****** **************************** Bottom of Data ****************************
```

What to do next

After reviewing the batch job, you can save it by using the SAVE command.

For information on what to expect when you run the batch job and what to do after the batch job completes, see “Analyzing a multi-target change” on page 801

Running a multi-target change

You can run all multi-target changes that are registered on a central system by using the Change Management (CM) batch interface. Running changes in batch is a more efficient alternative to analyzing changes one at a time through the CM panels.

About this task

Use the following procedure to manually define a batch job for running a multi-target change. As an alternative to this manual procedure, you can use panels to create a CM batch run job; for instructions, see "Running a multi-target change in batch by using panels" on page 807.

Procedure

To run a multi-target change:

1. In the CM batch file, specify ACTION_RUN_CHANGE = 'Y', as shown in the following example:

```
<JOB CARDS>
//*
//ADBLIBS JCLLIB ORDER=<CM batch PROCLIB>
//*
//***********************************************************GOCCM*******
//* STEP RUN A CHANGE
//***********************************************************************
//ANLYZ EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS  DD *
  ACTION_IMPORT_CHANGE='N'
  ACTION_ANALYZE_CHANGE='N'
  ACTION_RUN_CHANGE='Y'
  CHANGE_OWNER='SYSADM'
  CHANGE_NAME='CENTRAL MTC CHANGE - FOR TARGETS A & B'
 /*
```

   You can specify additional CM batch parameters as needed.

   Tip: If the multi-target change exists on another subsystem, use the MTC_CENTRAL_LOCATION parameter to specify the location of the change and use the CHANGE_NAME parameter to specify the name of the change at the central location. If you do not specify a central location, you must create a run job with a unique change name for each target at each remote location.

   Restriction: Currently, the DO_RUNTIME_ANALYZE parameter is not supported when running a multi-target change

2. Run the batch job.

   After the batch job runs successfully, a multi-target run report is generated, as shown in the following example:

```
--------------------------------------------------------------------------------
ADBCCM - Multi-target Change Summary Report (Run Process)
--------------------------------------------------------------------------------

Multi-target change id: 7913

Successful entries:

Location: DSNA

Target     ID     Owner    Name                         Status   Remarks
---------- ------ -------- ---------------------------- -------- ---------------
TARGET A1  7914   SYSADM   CHANGE1 FOR TARGET A1        COMPLETE Run Successful
TARGET A2  7915   SYSADM   CHANGE2 FOR TARGET A2        COMPLETE Run Successful


Skipped or Failed or Running entries:

Target     Loc    ID     Owner    Name                 Status   Remarks
---------- ------ ------ -------- --------------------- -------- ---------------

TARGET B   DSNB   N/A    SYSADM   CHANGE3 FOR TARGET B  ANALYZED Skipped. Target is not
                                                                 on local subsystem.
```

```
--------------------------------------------------------------------------------
ADBCCM - End of Multi-target Change Summary Report (Run Process)
--------------------------------------------------------------------------------
```

In this example, TARGET B was skipped because it is not on the local subsystem, DSNA. You must use subsystem DSNB to run the change for TARGET B.

What to do next

After running your changes, you might want to verify that your run was successful by completing the following steps:

1. On the **Change Management (CM) (ADB2C)** panel, specify option 1 (Manage changes), and press Enter.
2. On the **Manage Changes (ADB2C1)** panel, specify option 1 (Display changes), and press Enter.
3. On the**Changes (ADB2C11)** panel, issue the AT line command next to the multi-target change, and press Enter.

   The **CM - Associate Targets (ADBPCMT)** panel displays the status of all your target changes, as shown in the following example:

```
ADBPCMT n --------------- CM - Associate Targets ------------- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Details for multi-target change: SYSADM.CENTRAL MTC   > DB2 System: DSNA
                                                        DB2 SQL ID: SYSADM
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
    *        *             *         *                      *
--- -------> ------------> --------- ---------------------- -------------------
    TARGET A DSNA          SYSADM    CHANGE1 FOR TARGET A1  COMPLETE
    TARGET A DSNA          SYSADM    CHANGE2 FOR TARGET A2  COMPLETE
    TARGET B DSNB          SYSADM    CHANGE3 FOR TARGET B   DEFINED
******************************* END OF DB2 DATA *******************************
```

Figure 374. ***CM - Associate Targets (ADBPCMT)*** *panel*

Running a multi-target change in batch by using panels

You can use panels to create a Change Management (CM) batch job to run changes for multiple targets. Using a CM batch job can be more efficient than running changes one at a time with the CM panels.

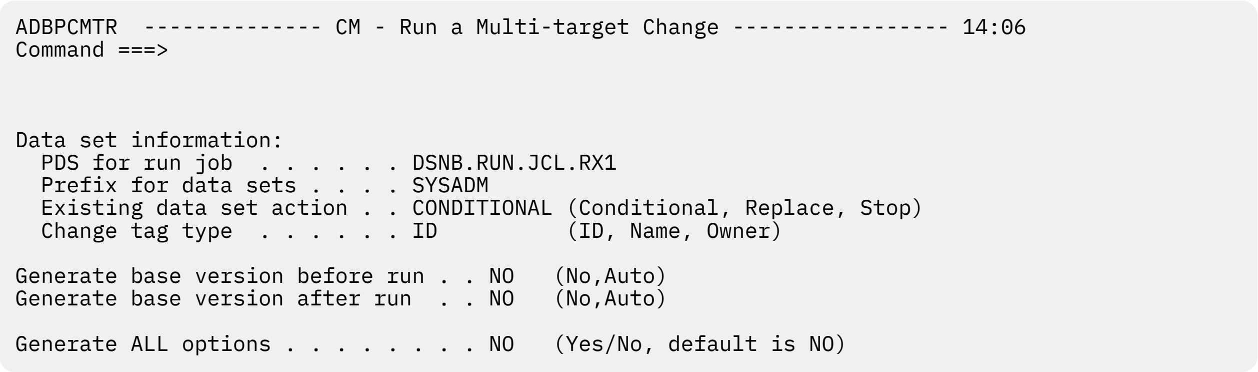
Procedure

To use panels to create a batch job to run changes for multiple targets:

1. On the **Change Management (CM) (ADB2C)** panel, specify option 1 (Manage changes), and press Enter.
2. On the **Manage Changes (ADB2C1)** panel, specify option 1 (Display changes), and press Enter.
3. On the **Changes (ADB2C11)** panel, issue one of the following line commands next to the change that you want to run:

   RN
   Creates a simple CM batch job without using any panel options.

   RNO
   Allows you to specify run options to include in the generated CM job. RNO can be specified only if the change type is MULTI-TC
4. On the **CM - Run a Multi-target Change (ADBPCMTR)** panel, complete the fields, and press Enter:



```
ADBPCMTR  -------------- CM - Run a Multi-target Change ----------------- 14:06
Command ===>



Data set information:
  PDS for run job  . . . . . . DSNB.RUN.JCL.RX1
  Prefix for data sets . . . . SYSADM
  Existing data set action . . CONDITIONAL (Conditional, Replace, Stop)
  Change tag type  . . . . . . ID          (ID, Name, Owner)

Generate base version before run . . NO   (No,Auto)
Generate base version after run  . . NO   (No,Auto)

Generate ALL options . . . . . . . . NO   (Yes/No, default is NO)
```

Figure 375. ***CM - Run a Multi-target Change (ADBPCMTR)*** *panel*

The first time this panel is used in a session, values are pre-filled with the CM batch default parameter values. For subsequent uses of this panel in a session, the previously entered values are reused. To reset the panel to use the default values, issue the DEFAULTS command. This command overrides any previously entered values.

Most fields on the panel correspond to CM batch parameters; see "CM batch parameter definitions" on page 664.

Batch parameters are generated in the job only if the corresponding value entered on the panel is different from the CM batch default value. If you want to generate CM batch options for all values that are specified on the panel, regardless of whether the values differ from the default values, set the **GENERATE ALL options** field to Y.

Note: If a target or targets exist on other subsystems, the generated jobs include additional steps for remote subsystems with the MTC_CENTRAL_LOCATION parameter and the appropriate location name of the change.

Depending on the values that you specify, you might be prompted for additional information before the run job is generated and the ISPF Edit session is displayed.

Results

A CM batch job is generated in the data set that is specified in the **PDS for run job** field, as shown in the following example:

```
****** ***************************** Top of Data ******************************
000001 //RHP1 JOB ,
000002 //*       RESTART=STEPNAME, <== FOR RESTART REMOVE * AND ENTER STEP NAME
000003 //     REGION=0M,NOTIFY=&SYSUID,
000004 //     MSGCLASS=H,TIME=(,30)
000005 //*
000006 /*JOBPARM S=SY4A
000007 //*
000008 //ADBLIBS JCLLIB ORDER=ADB.DEVCUST.PROCLIB
000009 //*
000010 //******************************************************GOCCM***********
000011 //* STEP REGISTER A CHANGE
000012 //*********************************************************************
000013 //GOCCM EXEC GOCCM,SSID=DSNA,PLAN=ADBDEV,GRP=VB2APAR
000014 //GOCCM.PARMS  DD *
000015  ACTION_RUN_CHANGE='Y'
000016  CHANGE_OWNER='SYSADM'
000017  CHANGE_NAME='TESTRHP_CHG03'
000018
000019  PREFIX_FOR_DATA_SETS = ’SYSADM’
000020  PDS_FOR_RUN_JCL = ’DSNB.RUN.JCL’
000021  EXISTING_DATA_SET_ACTION=’REPLACE’
000022  CHGTAG_TYPE = 'NAME'
000023
000024  GENERATE_BASE_VERSION_BEFORE_RUN = ’AUTO’
000025  GENERATE_BASE_VERSION_AFTER_RUN = ’AUTO’
000026  AUTH_SWITCH_USERID = 'SYSADM'
000027  AUTH_SWITCH_SECADM = 'SYSADM'
000028
```

```
000029  ;
****** **************************** Bottom of Data ****************************
```

What to do next

After reviewing the batch job, you can save it by using the SAVE command.

For information on what to expect when you run the batch job and what to do after the batch job completes, see "Running a multi-target change" on page 806

Using Db2 templates: Change Management (CM) batch interface

Managing templates when using the Change Management (CM) batch interface is done by specifying the Db2 TEMPLATE statement in `ADBTEMPL DD`. This enables installations to define a data set with Db2 TEMPLATE statements and to easily use these template statements in multiple Db2 subsystems.

About this task

Some Change Management (CM) batch interface parameters control whether user-provided templates or product default templates are used while others enable templates for utility type files.

Procedure

In the `ADBTEMPL` file, define each template on a separate line.

Depending on how you want to use templates, use the procedure in one of the following options.

Important: The first two words of a template statement must be TEMPLATE followed by the template name, with no SQL comments in between the first two words.

- To control whether user-provided templates or product default templates are used, use the following settings in the **`generate_templates`** parameter:
  - **Y**: If the `generate_templates` parameter is set to Y, the use of TEMPLATEs is enabled. If you enable templates for a template type individually (for example: util_template_copyddn1_use = 'S', user templates are used for that template type if it is defined in the ADBTEMPL file. If you do not enable templates for a template type, default templates are used for that template type.
  - **N**: If the `generate_templates` parameter is set to N, this allows you to easily disable the use of user-specified TEMPLATEs without having to toggle off or on each template type individually. When the `generate_templates` parameter is set to N, the results is that product default templates are used when templates are needed.
- To enable and make available templates for each utility file type, use the parameter names starting with **`util_template`** and **`util_clone_template`**.

  Note: The parameter names starting with **`util_clone_template`** define the templates used when processing a table space that has a clone table.

  For full descriptions of parameter names starting with **`util_template`** and **`util_clone_template`**, see "CM batch parameter definitions" on page 664.

Example

In the following example, templates COPY1 and LOBC are specified in the `ADBTEMPL DD`. The template named COPY1 is the product default template name for the first COPY data set. The template named LOBC is the product default template name for templates associated with LOB columns. The **`generate_templates`** parameter is set to Y, so these templates are used.

Note: The `ADBTEMPL` file is not processed to resolve product-specific and user-defined variables. The template statements are passed as is to Db2.

```
//LSCLIBS JCLLIB ORDER=DMTOOL.SGOCSAMP
//*
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB
```

```
//GOCCM.PARMS DD *
 CHANGE_NAME = 'AUTO:2011-12-11-15.24.28.803388'
ACTION_ANALYZE_CHANGE = 'Y'
generate_templates = 'Y'
take_an_image_copy = 'B'
run_reorg_rebuild = 'A'
run_check_data = 'Y'
prefix_for_data_sets = '&USERID..&ABC.'
;
 symbol_name = '&ABC.',symbol_value='TMPL';
/*
//ADBTEMPL DD *
TEMPLATE COPY1
  DSN 'DEMBIN2.TMPL.&SSID..COPY1.&UQ.'
TEMPLATE LOBC
  DSN 'DEMBIN2.TMPL.&SSID..LOB.&UQ.'
/*
```

- If the **generate_templates** parameter is set to N, the templates in ADBTEMPL DD are not used.
- If the **generate_templates** parameter is set to Y and the **util_template_copyddn1_name** parameter is set to ZZZ, the COPY1 template is not used for the first COPY data set because template ZZZ is not defined in the ADBTEMPL DD. In this case, a product default template is used. The LOBC template is still used whenever a template is needed for LOB columns.
- If the **generate_templates** parameter is set to Y, and **util_template_copyddn1_use** is set to ", the COPY1 template is not used for the first COPY data set because user-specified templates is disabled. The LOBC template is still used whenever a template is needed for LOB columns.

Examples: Invoking the Change Management (CM) batch interface for various actions

The following examples provide details about using the Change Management (CM) batch interface to performs various actions.

Important: For each of these examples, the PROFPARM file in the GOCCM JCL procedure contains the following parameter values:

```
JOB_PARM_LINE_1='S=SY4A'
JOB_JCLLIB_LINE_1='//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP';
```

- “Example 1: Importing a mask using the default mask name” on page 811
- “Example 2: Importing a mask using a user-provided mask name” on page 811
- “Example 3: Importing an ignore” on page 811
- “Example 4: Importing a DDL file using the default change name. The change is imported but not analyzed.” on page 812
- “Example 5: Importing more than one delta change file into a single change, and use a user provided change name. The change is imported but not analyzed.” on page 812
- “Example 6: Analyze a change.” on page 812
- “Example 7: Run a change.” on page 813
- “Example 8: Recover a change” on page 813
- “Example 9: Import, analyze, and build a run job in one invocation of CM batch” on page 814
- “Example 10: Import, analyze, build a run job, and run the change in one invocation of Change Management (CM) batch interface” on page 814
- “Example 11: Run compare and register a change to implement the differences” on page 815
- “Example 12: Run compare (same as example 11 but without registering a change)” on page 815
- “Example 13: Run compare, and do not register a change” on page 816
- “Example 14: Analyze a multi-target change” on page 816
- “Example 15: Run a multi-target change” on page 818

Example 1: Importing a mask using the default mask name

```
//IMMASK   JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//IMMASK   DD *
SGNAME:*,SYSDEFLT
/*
```

Once this job completes, a CM mask exists and is ready for use. The mask owner and name are something like:

```
MASK_OWNER = 'USER123'
MASK_NAME  = 'AUTO:2012-02-10-09.02.06.840242'
```

Example 2: Importing a mask using a user-provided mask name

```
//IMMASK   JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS  DD *
MASK_NAME = 'PROD_SCHEMA'
/*
//IMMASK DD *
SCHEMA:TEST*,PROD*
/*
```

Once this job completes, a CM mask exists and is ready for use. The mask owner and name are something like:

```
MASK_OWNER = 'USER123'
MASK_NAME  = 'PROD_SCHEMA'
```

Example 3: Importing an ignore

```
//IMIGNORE JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//IMIGNORE DD *
BPOOL
STGROUP
/*
```

Once this job completes, a CM ignore exists and is ready for use. The ignore owner and name are something like:

```
IGNORE_OWNER = 'USER123'
IGNORE_NAME  = 'AUTO:2012-02-10-09.02.06.840242'
```

Example 4: Importing a DDL file using the default change name. The change is imported but not analyzed.

```
//IMDDL    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *  ACTION_ANALYZE_CHANGE = 'N'
/*
//IMCHG001 DD *
-- FIRST LINE OF DDL FILE MUST BE A SIMPLE COMMENT!
CREATE TABLE IMPORT_DDL_DEMO1 (C1 INT);
CREATE TABLE IMPORT_DDL_DEMO2 (C1 INT);
/*
```

Once this job completes, a CM change exists and is ready for analyze. The change owner and name are something like:

```
CHANGE_OWNER = 'USER123'
CHANGE_NAME  = 'AUTO:2012-02-10-09.02.06.840242'
```

Example 5: Importing more than one delta change file into a single change, and use a user provided change name. The change is imported but not analyzed.

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
CHANGE_NAME = 'W023:&CURTS.'
CHANGE_COMMENT = 'THIS CHANGE IS FOR WORK ITEM W023.'
ACTION_ANALYZE_CHANGE = 'N'
/*
//IMCHG001 DD DISP=SHR,DSN=USER123.CMDEMOB.W001.DCHG
//IMCHG002 DD DISP=SHR,DSN=USER123.CMDEMOB.W002.DCHG
```

Tip: Instead of hard coding the work order number W023 in multiple places, use a user-defined symbol variable like the following.

```
//PARMS    DD *
 CHANGE_NAME = '&WORK#.:&CURTS.'
CHANGE_COMMENT = 'THIS CHANGE IS FOR WORK ITEM &WORK#..'
ACTION_ANALYZE_CHANGE = 'N'
symbol_name = '&WORK#.',
symbol_value = 'W023';
/*
```

Once this job completes, a CM change exists and is ready for analyze. The change owner and name are something like:

```
CHANGE_OWNER = 'USER123'
CHANGE_NAME  = 'W023:2012-02-10-09.25.43.232422'
```

Example 6: Analyze a change.

```
//ANCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
```

```
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
CHANGE_OWNER='USER123'
CHANGE_NAME='W023:2012-02-10-09.25.43.232422'
ACTION_ANALYZE_CHANGE = 'Y'
/*
```

Tip: The **`change_owner`** and **`change_name`** parameters were manually copied from the job output that imported the change. Here is an example snippet of the job output:

```
===========================
Detailed change information
===========================
For convenience, the change owner and name are displayed below using
the change management batch parameter syntax:
CHANGE_OWNER='USER123'
CHANGE_NAME='W023:2012-02-10-09.25.43.232422'
```

Once this job completes, the change is in 'ANALYZED' state and ready to be run.

Example 7: Run a change.

To run a change, submit the run job that was generated by Change Management (CM) batch interface. View the job output that analyzed the change to determine the location of the run job. For example, the run job location is listed for 'Run job DSN':

```

===========================
Detailed change information
===========================
For convenience, the change owner and name are displayed below using
the change management batch parameter syntax:
CHANGE_OWNER='USER123'
CHANGE_NAME='W023:2012-02-10-09.25.43.232422'

Change ID  . . . : 3075
Status . . . . . : ANALYZED
Created by . . . : USER123
Created  . . . . : 2012-02-10-09.25.44.796997
Last altered by  : USER123
Last altered . . : 2012-02-10-09.29.20.253278
Change type  . . : CHANGE
WSL DSN  . . . . : 'USER123.DSNA.ANALYZE.WSL(C0003075)'
Run job DSN  . . : 'USER123.DSNA.RUN.JCL(E0003075)'
Recover job DSN  : 'USER123.DSNA.RUN.JCL(R0003075)'
```

Submit the `'USER123.DSNA.RUN.JCL(E0003075)'` job to run the change. Once this job completes, the change is 'COMPLETE' which means the change was applied to Db2.

Example 8: Recover a change

To recover a change, submit the recover job that was generated by Change Management (CM) batch interface. View the job output that analyzed or ran the change to determine the location of the recover job. For example, the recover job location is listed for 'Recover job DSN':

```
===========================
Detailed change information
===========================
For convenience, the change owner and name are displayed below using
the change management batch parameter syntax:
CHANGE_OWNER='USER123'
CHANGE_NAME='W023:2012-02-10-09.25.43.232422'

Change ID  . . . : 3075
Status . . . . . : ANALYZED
Created by . . . : USER123
Created  . . . . : 2012-02-10-09.25.44.796997
Last altered by  : USER123
Last altered . . : 2012-02-10-09.29.20.253278
Change type  . . : CHANGE
```

```
WSL DSN  . . . . : 'USER123.DSNA.ANALYZE.WSL(C0003075)'
Run job DSN  . . : 'USER123.DSNA.RUN.JCL(E0003075)'
Recover job DSN  : 'USER123.DSNA.RUN.JCL(R0003075)'
```

Submit the `'USER123.DSNA.RUN.JCL(R0003075)'` job to recover the change. Once this job completes, the change is recovered. The change status is set back to 'DEFINED'.

Example 9: Import, analyze, and build a run job in one invocation of CM batch

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
/*
//IMCHG001 DD *
-- FIRST LINE OF DDL FILE MUST BE A SIMPLE COMMENT!
 CREATE TABLE IMPORT_DDL_DEMO3 (C1 INT);
/*
//IMCHG002 DD *
-- FIRST LINE OF DDL FILE MUST BE A SIMPLE COMMENT!
ALTER  TABLE IMPORT_DDL_DEMO3
 ADD COLUMN C2 INT;
/*
```

Note: A delta change file could have been specified for the `IMCHG001` and `IMCHG002` files instead of specifying DDL.

Once this job completes, a CM change exists and is ready to run. The change status is 'ANALYZED'. The change owner and name are something like:

```
CHANGE_OWNER = 'USER123'
CHANGE_NAME  = 'AUTO:2012-02-10-09.26.33.236111'
```

Example 10: Import, analyze, build a run job, and run the change in one invocation of Change Management (CM) batch interface

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
ACTION_RUN_CHANGE = 'Y'
/*
//IMCHG001 DD *
-- FIRST LINE OF DDL FILE MUST BE A SIMPLE COMMENT!
CREATE TABLE IMPORT_DDL_DEMO4 (C1 INT);
/*
//IMCHG002 DD *
-- FIRST LINE OF DDL FILE MUST BE A SIMPLE COMMENT!
ALTER  TABLE IMPORT_DDL_DEMO4
 ADD COLUMN C2 INT;
/*
```

Note: A delta change file could have been specified for the `IMCHG001` and `IMCHG002` files instead of specifying DDL.

Once this job completes, a CM change exists and is applied to Db2. The change status is 'COMPLETE'. The change owner and name are something like:

```
CHANGE_OWNER = 'USER123'
CHANGE_NAME  = 'AUTO:2012-02-10-09.26.36.636543'
```

Example 11: Run compare and register a change to implement the differences

The compare source is DDL and the compare target is from the Db2 catalog where the Db2 objects are automatically selected based on the content of the source.

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//* INSERT NEW COLUMN NEWCOL INTO TABLE CMBSAMP.TB01
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
   ACTION_COMPARE = 'Y'
/*
//SRCIN DD *
SET CURRENT SQLID = 'DEMBIN2';
CREATE DATABASE CMBSAMP;
COMMIT;
CREATE TABLESPACE CMBSAMP IN CMBSAMP
   MAXPARTITIONS 10;
COMMIT;
CREATE TABLE CMBSAMP.TB01
(C1 INT NOT NULL WITH DEFAULT
,NEWCOL  INT NOT NULL WITH DEFAULT
,C3 INT NOT NULL WITH DEFAULT)
IN CMBSAMP.CMBSAMP;
CREATE INDEX CMBSAMP.TB01IX01
  ON CMBSAMP.TB01 (C1);
CREATE VIEW CMBSAMP.VW01 (C1,C3) AS
  SELECT C1,C3 FROM CMBSAMP.TB01;
/*
```

The job output contains the compare report, and message ADB9917I that lists the location of the output version files and of the Db2 Admin Tool delta change file.

```
ADB9917I Compare data set information:
           Delta change data set name:
             DSN=DEMBIN2.SAMP11.OC.D2013127.T132255.DELTA

           Source version:
             Type  . . : FILE
             Owner . . :
             Name  . . : DEMBIN2.SAMP11.OC.D2013127.T132255.SRCVF

           Target version:
             Type  . . : FILE
             Owner . . :
             Name  . . : DEMBIN2.SAMP11.OC.D2013127.T132255.TGTVF
```

Example 12: Run compare (same as example 11 but without registering a change)

Set **action_import_change** = 'N'.

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//* INSERT NEW COLUMN NEWCOL INTO TABLE CMBSAMP.TB01
```

```
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
  ACTION_COMPARE = 'Y'
  ACTION_IMPORT_CHANGE = 'N'
/*
//SRCIN DD *
SET CURRENT SQLID = 'DEMBIN2';
CREATE DATABASE CMBSAMP;
COMMIT;
CREATE TABLESPACE CMBSAMP IN CMBSAMP
  MAXPARTITIONS 10;
COMMIT;
CREATE TABLE CMBSAMP.TB01
(C1 INT NOT NULL WITH DEFAULT
,NEWCOL  INT NOT NULL WITH DEFAULT
,C3 INT NOT NULL WITH DEFAULT)
IN CMBSAMP.CMBSAMP;
CREATE INDEX CMBSAMP.TB01IX01
  ON CMBSAMP.TB01 (C1);
CREATE VIEW CMBSAMP.VW01 (C1,C3) AS
  SELECT C1,C3 FROM CMBSAMP.TB01;
/*
```

The job output contains the compare report, and message ADB9917I as described in example 11.

Example 13: Run compare, and do not register a change

The compare source and target is a user-provided list of Db2 object names, and masking is specified.

```
//IMCHG    JOB (&SYSUID),'DEMO',CLASS=A,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH  EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS    DD *
  ACTION_COMPARE = 'Y'
  ACTION_IMPORT_CHANGE = 'N'
  SOURCE_TYPE = 'USER'
  TARGET_TYPE = 'USER'
/*
//SRCIN DD *
TYPE='DB' NAME='DBTV2';
//TGTIN DD *
TYPE='DB' NAME='DBTV1';
//MASKS DD *
DBNAME:DBTV2,DBTV1
SCHEMA:SCTV2,SCTV1
/*
```

The job output contains the compare report, and message ADB9917I as described in example 11.

Example 14: Analyze a multi-target change

```
//IMMASK JOB (&SYSUID),’DEMO’,CLASS=A,
// MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
// REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS DD *
ACTION_IMPORT_CHANGE='N'
ACTION_ANALYZE_CHANGE='Y'
ACTION_RUN_CHANGE='N'
EXISTING_DATA_SET_ACTION='REPLACE'
DO_RUNTIME_ANALYZE='N'
REPORT_OBJECT_COUNT='Y'
REPORT_OBJECT_TOTALS='Y'
REPORT_SUMMARY = 'Y'
```

```
CHANGE_OWNER='MATTV'
CHANGE_NAME='PSV_GROUP A'
CHANGE_COMMENT=''
;
/*EOF
```

The CM batch job produces a report that contains data based on the reporting options specified on the PARMS DD statement. In this example, the following sections are included in the REPORT file:

Comparison Summary Report

Specify REPORT_SUMMARY = 'Y' to indicate that you want to produce a Summary Report. The summary report contains one line for each object that was compared and the result of the comparison. Long authorization IDs or names cannot be shown on a single line and are truncated. For long authorization IDs, the first eight characters are shown, followed by a > to indicate a long authorization ID. For long object names, the first 18 characters are shown, followed by a > to indicate a long name.

Comparison Counts Report

Specify REPORT_OBJECT_COUNT='Y' to indicate that you want the report to include a Comparison Counts 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. There is one report for each multi-target change target.

The following information is provided in the Comparison Counts Report:

- Objects that are not found in the target are reported as ADDED. There are various reasons why some objects cannot be added to the target; these objects are reported in the NOT ADDED column.
- Objects that are not found in the source are reported as DROPPED
- Objects that are modified through the ALTER statement are reported as ALTERED.
- Objects that failed to be added are reported in the NOT ADDED column. The reason an object was not added is explained in the Object Comparison Report.
- Objects that are modified through DROP and CREATE are reported as RECREATED.

Totals Report

Specify REPORT_OBJECT_TOTALS='Y' to indicate that you want the report to include a Totals Report. The Totals Report shows information similar to the Comparison Counts Report, as well as the sum of the counts from all of the multi-target change targets.

A sample Totals Report is shown in the following figure:

```
Totals Report
=============

Object Type              Alter      Create        Drop   Recreated      Totals
-------------------  ----------  ----------  ----------  ----------  ----------
Schemas                       0           0           0           0           0
User Def Types                0           0           0           0           0
Sequences                     0           0           0           0           0
Global Variables              0           0           0           0           0
Databases                     0           0           0           0           0
Tablespaces                   0           0           0           0           0
Tables                        3           0           0           0           3
Indexes                       0           0           0           0           0
Aliases                       0           0           0           0           0
Storage Groups                0           0           0           0           0
Synonyms                      0           0           0           0           0
Functions                     0           0           0           0           0
Stored Procedures             0           0           0           0           0
Triggers                      0           0           0           0           0
Views                         0           0           0           0           0
Column Masks                  0           0           0           0           0
Row Permissions               0           0           0           0           0
Relations (FKs)               0           0           0           0           0
Unique Constraints            0           0           0           0           0
-------------------  ----------  ----------  ----------  ----------  ----------
  Totals                      3           0           0           0           3
```

The following information is provided in the Totals Report:

Alter
Objects that are modified through the ALTER statement

Create
Objects that are not found in the target

Drop
Objects that are not found in the source

Recreated
Objects that are modified through DROP and CREATE statements

Though unique constraints are not Db2 objects, they are also included in the Totals Report:

Alter
There are no altered unique constraints

Create
Constraints that are added with a new table or constraints that are added on an existing table

Drop
Constraints that are dropped with a dropped table or constraints that are dropped from an existing table (if the constraint is on the target only)

Recreated
Constraints without any changes that are dropped and recreated because the table is dropped and recreated

Example 15: Run a multi-target change

```
//IMMASK JOB (&SYSUID),’DEMO’,CLASS=A,
// MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
// REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//CMBATCH EXEC GOCCM,SSID=DSNA,PLAN=ADB
//PARMS DD *
ACTION_IMPORT_CHANGE='N'
ACTION_ANALYZE_CHANGE='N'
ACTION_RUN_CHANGE='Y'
REPORT_STMT_TOTALS='Y'
CHANGE_OWNER='MATTV'
CHANGE_NAME='PSV_GROUP A'
EXISTING_DATA_SET_ACTION='REPLACE'
REPORT_SUMMARY = 'Y'
RUN_CHECK_DATA        = 'N'
RUN_REBIND            = 'M'
RUN_REORG_REBUILD     = 'M'
RUN_RUNSTATS          = 'N'
UNLOAD_METHOD         = 'U'
TAKE_AN_IMAGE_COPY    = 'N'
;
/*EOF
```

The CM batch job produces a report that contains data based on the reporting options specified on the PARMS DD statement. In this example, the following sections are included in the ADBRPTSM file:

Totals Report
Specify REPORT_STMT_TOTALS='Y' to indicate that you want the report to include a Totals Report. The Totals Report shows information for each ALTER statement, each CREATE statement, and each DROP statement that is issued, per object type. There is one Totals Report for each target system and then an overall report that includes the sum of all of the targets.

Note: The following conditions apply for totals reports:

- Implicit impact is not considered. Dropping one database may implicitly drop many other objects that are not counted.
- If multiple statements are issued for the same object, those statements are counted separately.

A sample Totals Report is shown in the following figure:

```
Totals Report: Target C0004708
 ==============================
 ADB9963I This report displays totals for all of the statements issued.

 Object Type               Alter      Create        Drop      Totals
 -------------------  ----------  ----------  ----------  ----------
 Schemas                       0           0           0           0
 User Def Types                0           0           0           0
 Sequences                     0           0           0           0
 Global Variables              0           0           0           0
 Databases                     0           0           0           0
 Tablespaces                   0           0           0           0
 Tables                        1           0           0           1
 Indexes                       0           0           0           0
 Aliases                       0           0           0           0
 Storage Groups                0           0           0           0
 Synonyms                      0           0           0           0
 Functions                     0           0           0           0
 Stored Procedures             0           0           0           0
 Triggers                      0           0           0           0
 Views                         0           0           0           0
 Column Masks                  0           0           0           0
 Row Permissions               0           0           0           0
 Relations (FKs)               0           0           0           0
 Unique Constraints            0           0           0           0
 -------------------  ----------  ----------  ----------  ----------
   Totals                      1           0           0           1
```

```
Totals Report: Overall
 ======================
 ADB9963I This report displays totals for all of the statements issued.

 Object Type               Alter      Create        Drop      Totals
 -------------------  ----------  ----------  ----------  ----------
 Schemas                       0           0           0           0
 User Def Types                0           0           0           0
 Sequences                     0           0           0           0
 Global Variables              0           0           0           0
 Databases                     0           0           0           0
 Tablespaces                   0           0           0           0
 Tables                        3           0           0           3
 Indexes                       0           0           0           0
 Aliases                       0           0           0           0
 Storage Groups                0           0           0           0
 Synonyms                      0           0           0           0
 Functions                     0           0           0           0
 Stored Procedures             0           0           0           0
 Triggers                      0           0           0           0
 Views                         0           0           0           0
 Column Masks                  0           0           0           0
 Row Permissions               0           0           0           0
 Relations (FKs)               0           0           0           0
 Unique Constraints            0           0           0           0
 -------------------  ----------  ----------  ----------  ----------
   Totals                      3           0           0           3
```

The following information is provided in the Totals Report:

Alter
The number of ALTER statements issued

Create
The number of CREATE statements issued

Drop
The number of DROP statements issued

Though unique constraints are not Db2 objects, they are also included in the Totals Report:

Alter
There are no altered unique constraints

Create
Constraints that are added with a new table or constraints that are added on an existing table

Drop

Constraints that are dropped with a dropped table or constraints that are dropped from an existing table (if the constraint is on the target only)

Example 16. Specifying where to restart work statement lists

```
ACTION_RUN_CHANGE = 'Y'
DO_RUNTIME_ANALYZE = 'N'
CHANGE_NAME = 'TEST1'
CHANGE_COMMENT = 'CHANGE_1'

ADBTEP2_RESTART='U'
ADBTEP2_USER_RESTART_NAME='MIRAGE'

ADBTEP2_RESTART_ENV='BOTH'
ADBTEP2_RESTART_REPORT_ONLY='N'
ADBTEP2_RESTART_SQLID='TS5771'
ADBTEP2_RESTART_SCHEMA='TS5771'
ADBTEP2_RESTART_SERVER=' '
ADBTEP2_RESTART_PATH=' '
ADBTEP2_RESTART_PRECISION='5'
ADBTEP2_RESTART_RULES='DB2'
ADBTEP2_RESTART_DECFLOAT_ROUNDING_MODE='ROUND_CEILING'
ADBTEP2_RESTART_ROUTINE_VERSION='1'
ADBTEP2_RESTART_BUSINESS_TIME=' '
ADBTEP2_RESTART_SYSTEM_TIME=' '
ADBTEP2_RESTART_GETARCHIVE=' '
ADBTEP2_RESTART_MOVETOARCHIVE=' '
ADBTEP2_RESTART_UNLLOBXML=' '
ADBTEP2_RESTART_UNLNOIC=' '
```

Example 17. Tracking and running non-Change-Management work statement lists by using CM Batch

```
ACTION_COMPARE = 'Y'
ACTION_GENERATE_WSL = 'Y'
ACTION_IMPORT_CHANGE = 'N'
WORKLIST_NAME = 'worklist_name'
PDS_FOR_WSL = 'pds_name'
```

Change Management scenarios

Change Management scenarios illustrate how you might use Change Management to make a simple change to a database and move changes that are made on one system to another.

Topics:

- "Scenario: Making a simple change to a database" on page 820
- "Scenario: Promoting changes from one system to another" on page 822

Scenario: Making a simple change to a database

This scenario explains how to make changes to part of a database structure on a development system.

About this task

Specifically, for the EMP table, you want to drop the COMMISSION column and increase the length of the LASTNME column to 45 bytes.

In making these changes, you have the following goals:

- Ensure that there is a snapshot of the database structure for fallback purposes.
- For the dropped column, repair any side effects of the change, such as handling inoperative or undefined objects such as packages, views, and triggers.
- Ensure that data is preserved for the change to the column length.

- Optimize the database with respect to the changes, such as running RUNSTATS or rebinding where necessary.
- Capture the changes for auditing purposes.

The following steps show you how you might use Change Management to make these changes and achieve your goals:

Procedure

1. Generate operations to track the change in Change Management.

   Create a version scope of the human resources database. You want to define a version scope because you want to create a snapshot (or base version) of the database structure after the changes are made. The version scope defines the objects that should be in the base version.
2. Modify the length of the LASTNME column and drop the COMMISSION column.
   a) Find and select the EMP table.
   b) Issue the ALT command to change the table. If there are any pending changes to the table, specify whether to implement your changes based on the assumption that the pending changes have been performed or that they have not been made and your change should supersede them. In this scenario, assume that there are no pending changes.
   c) Type over the length of the LASTNME column to increase the length to 45.
   d) Issue the D line command to delete the COMMISSION column.
3. Identify the impact that the changes have. To assess the impact of increasing the length of the LASTNME column and dropping the COMMISSION column:
   a) Type the REL primary command to see the related objects.
   b) Select each related object individually and determine if any changes are required because of the change in length to LASTNME or for COMMISSION being dropped. In this example, assume that a view is impacted by the dropped column.
4. Repair the side effects for the change. To fix the view:
   a) Issue the A line command to change the view.
   b) In the edit session that is displayed, remove the predicate from the view and save the edit session. The new definition of the view will be included as part of the change.
   c) Issue the CONTINUE command to finalize the changes to the table and the view.
5. Register the change in the Change Management database.

   Change registration occurs in this scenario because Change Management is enabled and required. To register the change, specify an owner and name for the change.
6. Analyze the change. To analyze the change:
   a) Go to the Change Management main menu and display the list of changes.
   b) Issue the analyze command for the change.
   c) Submit the batch job that Db2 generates to perform the analyze.

      The batch job produces a report of the changes that will be made and generates a WSL that will make the changes.
7. Run the change and capture a snapshot of the database structure after the change is complete. To run the change:
   a) Go to the Change Management main menu and display the list of changes.
   b) Issue the run command for the change, specifying that a new base version of the database structure should be created after the changes are made.
   c) Submit the batch job that runs the WSL that applies the changes.

Scenario: Promoting changes from one system to another

This scenario supposes that you are asked to promote the changes that were made to the human resources database on the development system to the test system.

About this task

This task requires you to determine the differences between the development and test system and apply the changes to the test system.

Assume that versions for the current state of the databases exist (Release 11C in DEV and Release 11B in TEST). You will compare the two versions to generate a delta changes data set that contains the SQL statements that represent the differences, transfer the delta changes data set to the test system, import the delta changes data set on the test system as a new change, and then apply the changes to the test database.

In synchronizing the human resources database, you have the following goals:

- Ensure that there is a snapshot of the database structures for fallback purposes.
- Capture the changes that are made on the test system for auditing purposes.

The following steps show you how you might use Change Management to make these changes and achieve your goals:

Procedure

1. On the source system (the development system), use the current versions of the development and test databases to identify the differences between the databases and promote the differences to the target system (the test system). To find and promote the differences in a delta changes data set:
   a) Go to the Change Management main menu and display the Manage Changes panel.
   b) Select the option to create a delta changes file for the target system.
   c) Identify the version of the test database as the starting version and the version of the development database as the ending version. Provide a name for the job that will generate the delta changes data set and a name for the delta changes data set.

      The starting version is a snapshot of the objects before changes are made, and the ending version is a snapshot of the objects after changes are made. In this scenario, you want to bring the level of the test system up to the level of the development system.
   d) Register the change in the Change Management database.

      You will be prompted to register the changes that are being promoted.
   e) Submit the batch job that creates the delta changes data set.
2. Import the delta changes data set as a new change on the test system. To import the promoted changes on the test system:
   a) Go the Change Management main menu on the test system and display the **Manage Changes** panel.
   b) Select the option to import changes.
   c) Specify the name of the delta changes data set to import into a change, and register the imported change in the Change Management database.

      Importing a change is a two-step process. First, Db2 Admin Tool performs an analysis to determine if there are any prerequisite changes that are pending for the objects that are affected by the imported change. Next, the change is registered. The steps can be performed either in the foreground (TSO) or the background (batch).
3. Analyze the imported change. To analyze the change:
   a) Go to the Change Management main menu and display the list of changes.
   b) Issue the analyze command for the change.
   c) Submit the batch job that Db2 generates to perform the analyze.

The batch job produces a report of the changes that will be made and generates a WSL that will make the changes.

4. Run the imported change and capture a snapshot of the test human resources database after the change is complete. To run the change:
   a) Go to the Change Management main menu and display the list of changes.
   b) Issue the run command for the change, specifying that a new base version of the database structure should be created after the changes are made.
   c) Submit the batch job that runs the WSL that applies the changes.

Multi-target changes

You can register a change to any catalog object on one system and import the change on multiple target systems. Changes are prepared on a central system and then applied to one or more target systems.

Before you begin

Admin Tool calls stored procedure ADBCRSP to update the change management database for multi-target changes. You should configure the WLM address space so that it has access to load modules ADBCRSP, ADB3000, and ADB9000 by copying these modules to a library defined in the STEPLIB concatenation for the WLM address space. The following example uses the ADMIN.WLM.LOAD load module data set:

```
000024 //STEPLIB  DD
DSN=ADMIN.WLM.LOAD,DISP=SHR
000025//          DD  DSN=USER.TESTLIB,DISP=SHR
000026 //         DD  DSN=DB2A.UTLIB,DISP=SHR
000027 //         DD  DSN=DB2A.TESTLIB,DISP=SHR
000028 //         DD  DSN=DB2A.SDSNLOAD,DISP=SHR
000029 //         DD  DSN=DB2A.SDSNLOD2,DISP=SHR
000030 //         DD  DSN=CEEA.SCEERUN,DISP=SHR
```

About this task

The following topics show you how you might deploy a change on multiple targets.

Setting up the targets

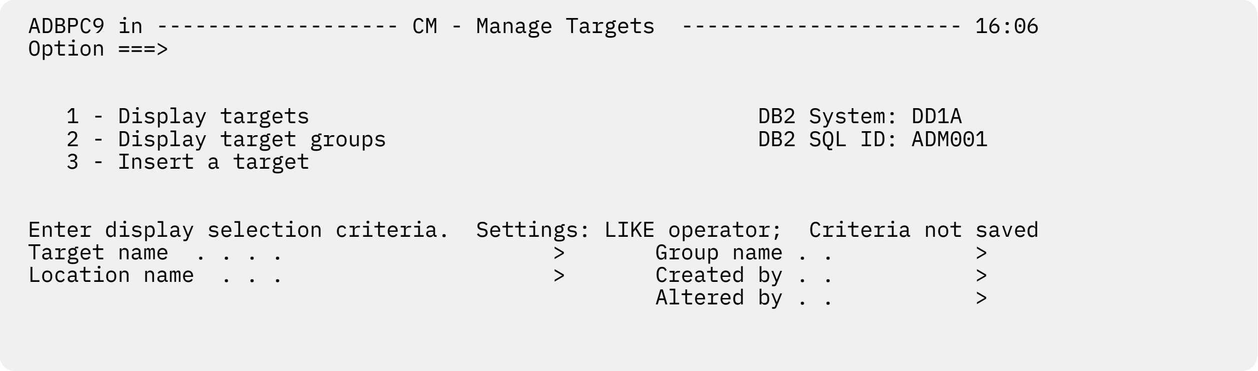
You can set up all the targets that you want to deploy changes to through the **Change Management** panel.

About this task

Procedure

1. Specify option 9, Manage targets on the **Change Management** panel.

   The CM - Manage Targets panel is displayed, as shown in the following figure:



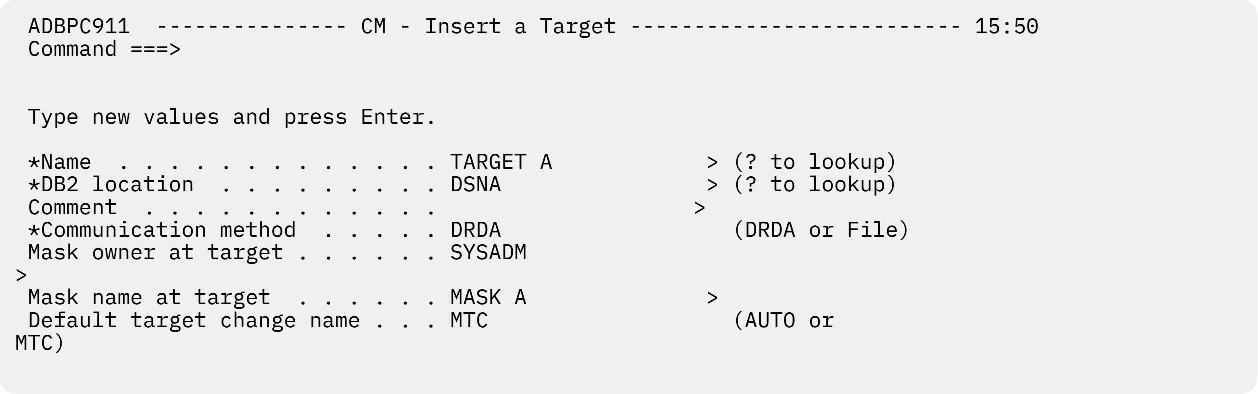
```
ADBPC9 in ------------------- CM - Manage Targets  ---------------------- 16:06
Option ===>


   1 - Display targets                                DB2 System: DD1A
   2 - Display target groups                          DB2 SQL ID: ADM001
   3 - Insert a target


Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Target name  . . . .                     >       Group name . .           >
Location name  . . .                     >       Created by . .           >
                                                 Altered by . .           >
```

Figure 376. CM - manage Targets (ADBPC9)

On the Manage Targets panel, you can display targets or create a target. When you use option 1 or 2, you can qualify the search by using the additional search criteria fields at the bottom of the screen.

2. Specify option 3, Insert a target on the **Manage Targets** panel.

   The Insert a Target panel is displayed, as shown in the following figure:



```
 ADBPC911  --------------- CM - Insert a Target ------------------------- 15:50
 Command ===>


 Type new values and press Enter.

 *Name  . . . . . . . . . . . . . TARGET A            > (? to lookup)
 *DB2 location  . . . . . . . . . DSNA                > (? to lookup)
 Comment  . . . . . . . . . . . .                    >
 *Communication method  . . . . . DRDA                  (DRDA or File)
 Mask owner at target . . . . . . SYSADM
>
 Mask name at target  . . . . . . MASK A              >
 Default target change name . . . MTC                   (AUTO or
MTC)
```

Figure 377. CM - Insert a target panel (ADBPC911)

The following fields are displayed on this panel:

Name
The name given for the target.

DB2 location
The location of the remote server.

Comment
An optional field you use to enter a comment to describe the target.

Communication method
Specify the method used to register changes to this target:

DRDA
Use DRDA when registering changes to this target.

FILE
Use the file method to register changes to this target. Note that a file is written with information for all targets regardless of which method is specified.

Mask owner at target
Specify the owner of the default mask that exists at the target location.

Mask name at target
Specify the name of the default mask that exists at the target location.

Default target change name
Specify the default format of the target change name. You can choose one of the following format options:

AUTO
AUTO:<timestamp>

MTC
<MTC change name>:<Target name>:<MTC location>

Note: For the MTC format, the combination of MTC change name, target name, and MTC location that you specify must be unique.

3. Set up the new target by specifying the details on the **CM - Insert a Target** panel and then press Enter.

   The target is inserted.

4. To add another target, repeat “3” on page 824 until all targets are configured.

Displaying targets

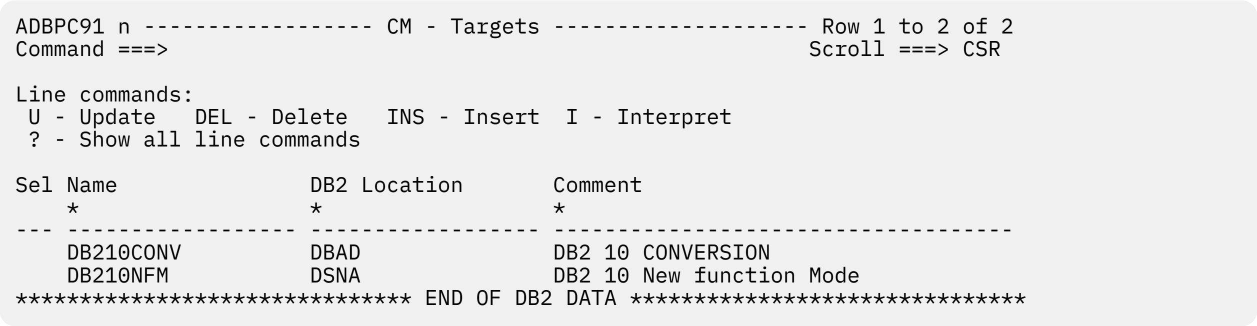
You can manage target profile definitions and specify selection criteria for displaying a list of target profiles.

About this task

You can create a multi-target change in which changes you make on a central system are propagated to one or more targets. A target is a Db2 subsystem where you wish to apply the change made on the central system.

Procedure

1. Specify option CM, Change Management on the **DB2 Administration Menu (ADB2)** panel.
2. Specify option 9, Manage targets on the Change Management panel.
3. Specify option 1, Display targets on the **CM - Manage Targets** panel. The following figure shows the **CM - Targets** panel.



```
ADBPC91 n ------------------ CM - Targets -------------------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update   DEL - Delete   INS - Insert  I - Interpret
 ? - Show all line commands

Sel Name               DB2 Location       Comment
    *                  *                  *
--- ------------------ ------------------ ------------------------------------
    DB210CONV          DBAD               DB2 10 CONVERSION
    DB210NFM           DSNA               DB2 10 New function Mode
******************************* END OF DB2 DATA *******************************
```

Figure 378. Manage Targets panel (ADBPC91)

4. Select one of the following line commands to work with the target.

U
Update the current target entry using the Insert a Target panel (ADBPC911).

DEL
Delete the current target entry.

INS
Insert a target panel (ADBPC911).

I
Provide an interpretation of the target. This option displays the name, Db2 location, comment, communication method, the mask name and owner at the target, the ID of the person who created the target, and the date it was last altered.

Displaying target groups

A target group is an optional entity that represents a set of target environments. You create a group name and select the targets that comprise the group. You can create or display target groups.

About this task

A target can be defined in one or more groups. Groups can be redefined as needed.

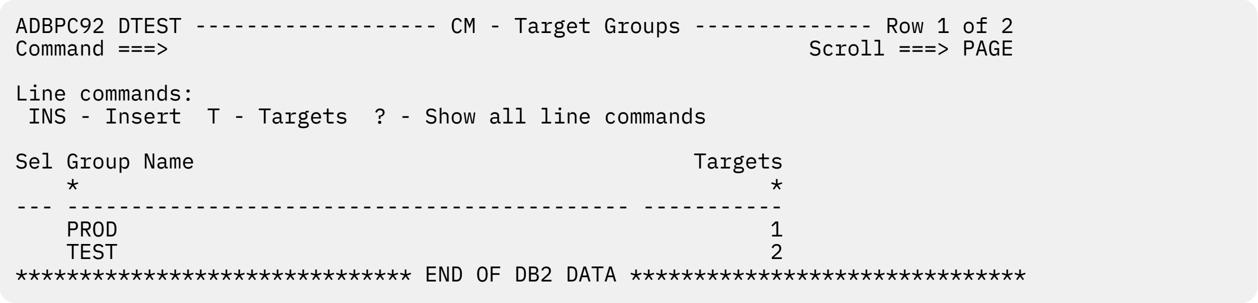
To display target groups:

Procedure

1. Specify option CM, Change Management on the **DB2 Administration Menu (ADB2)** panel.
2. Specify option 9, Manage targets on the Change Management panel.

3. Specify option 2, Display target groups, on the CM - Manage Targets panel.

   If no target groups exist, panel ADBPC921 is displayed, allowing you to insert a group. If a target group exists, panel ADBPC92 is displayed, as shown in the following figure:

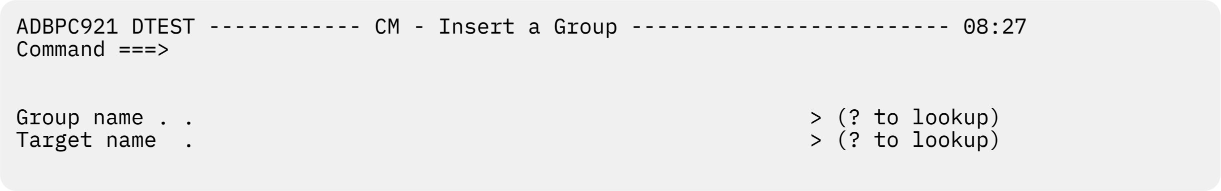


```
ADBPC92 DTEST ------------------- CM - Target Groups -------------- Row 1 of 2
Command ===>                                                  Scroll ===> PAGE

Line commands:
 INS - Insert  T - Targets  ? - Show all line commands

Sel Group Name                                       Targets
    *                                                      *
--- -------------------------------------------- -----------
    PROD                                                   1
    TEST                                                   2
******************************* END OF DB2 DATA *******************************
```

Figure 379. Manage Targets panel (ADBPC92)

4. Select one of the following line commands to work with the target group.

   - Selecting the INS line command displays panel ADBPC921, as shown in the following figure:

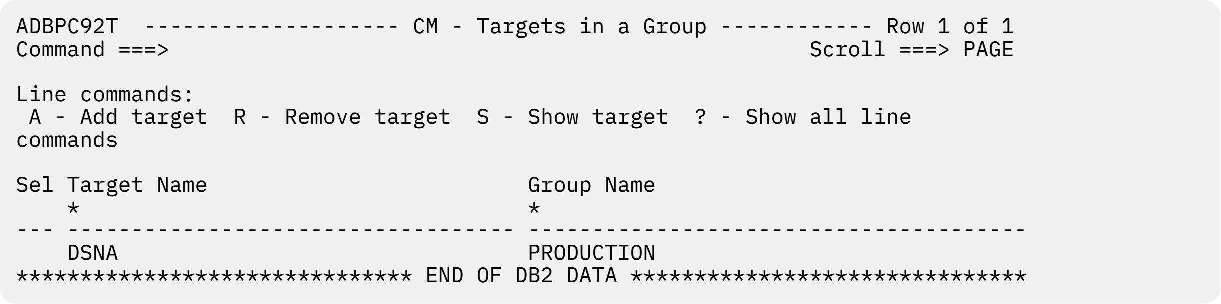


```
ADBPC921 DTEST ------------ CM - Insert a Group ------------------------- 08:27
Command ===>


Group name . .                                                > (? to lookup)
Target name  .                                                > (? to lookup)
```

Figure 380. Insert a Group panel (ADBPC921)

This panel allows you to insert a target group and target location entry. Enter the Group name to indicate the name of the target group, and the Target name to indicate the target name to include in the group.

   - Selecting the T command displays panel ADBPC92T, as shown in the following figure:



```
ADBPC92T  -------------------- CM - Targets in a Group ------------ Row 1 of 1
Command ===>                                                  Scroll ===> PAGE

Line commands:
 A - Add target  R - Remove target  S - Show target  ? - Show all line
commands

Sel Target Name                         Group Name
    *                                   *
--- ----------------------------------- ---------------------------------------
    DSNA                                PRODUCTION
******************************* END OF DB2 DATA *******************************
```

Figure 381. Targets in a Group panel (ADBPC92T)

The S line command displays panel ADBPC91. The R line command removes the target from the group. If it is the last target in the group, the group is removed.

Registering a multi-target change

You can register and track changes on multiple target systems.

Before you begin

Change Management must be enabled on the system and be either optional or required for your SQL ID. You enable Change Management by customizing the Db2 Administration Tool.

If the option to create a multi-target change is shown on the Register Options panel (ADB2CRO), then the change can be registered on multiple target systems. The steps that follow assume your system is configured to create a multi-target change.

You can register the change on multiple target locations.

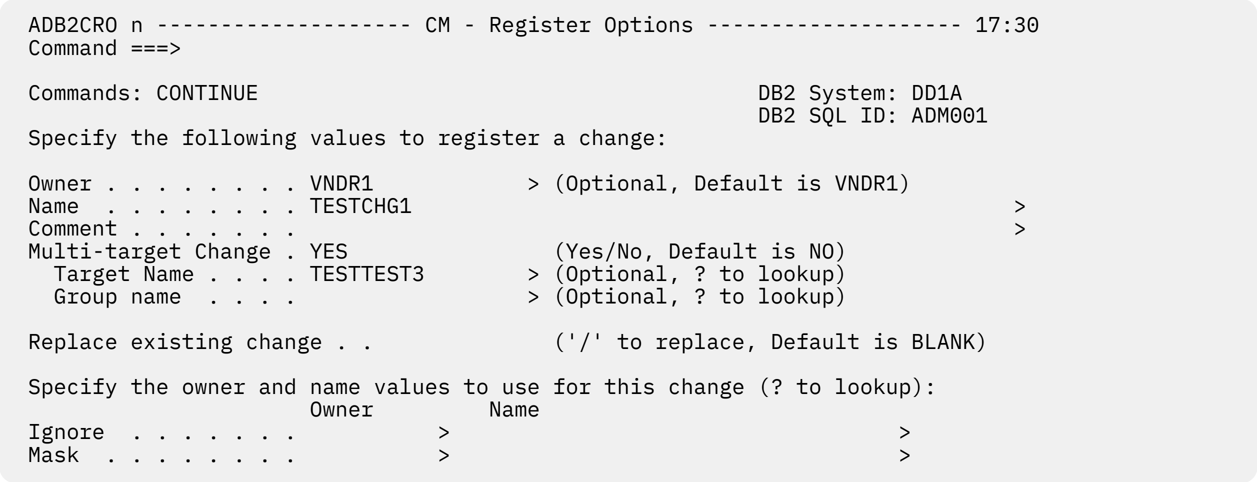
About this task

To register a multi-target change:

Procedure

1. Specify Yes in the Multi-target Change field on the **CM - Register Options** panel and then issue the CONTINUE command.

   The following figure shows an example of the CM - Register Options panel:

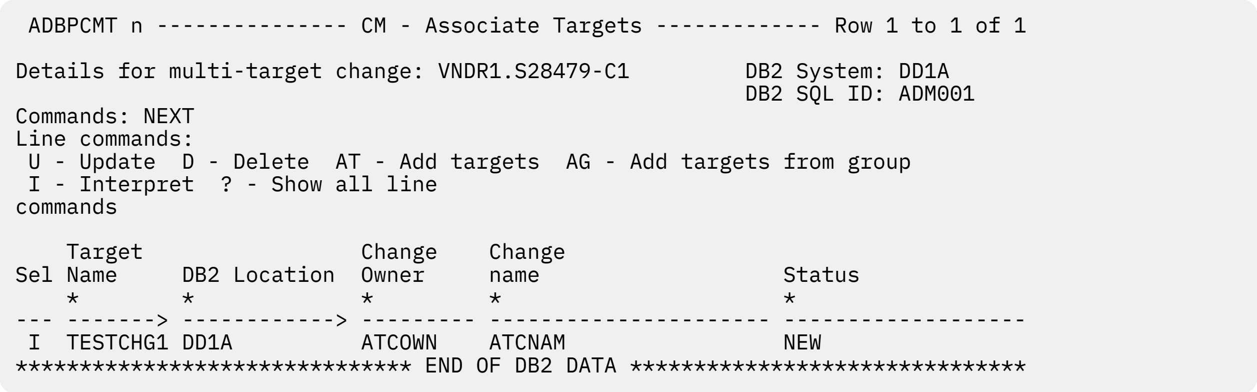


```
 ADB2CRO n -------------------- CM - Register Options -------------------- 17:30
 Command ===>

 Commands: CONTINUE                                       DB2 System: DD1A
                                                          DB2 SQL ID: ADM001
 Specify the following values to register a change:

 Owner . . . . . . . . VNDR1           > (Optional, Default is VNDR1)
 Name  . . . . . . . . TESTCHG1                                          >
 Comment . . . . . . .                                                   >
 Multi-target Change . YES               (Yes/No, Default is NO)
   Target Name . . . . TESTTEST3       > (Optional, ? to lookup)
   Group name  . . . .                 > (Optional, ? to lookup)

 Replace existing change . .             ('/' to replace, Default is BLANK)

 Specify the owner and name values to use for this change (? to lookup):
                       Owner         Name
 Ignore  . . . . . . .           >                                    >
 Mask  . . . . . . . .           >                                    >
```

Figure 382. CM - Register Options Panel (ADB2CRO)

2. Select the target names you want to register on the ADBPCMT panel then select NEXT.

   The following figure shows an example of the CM - Associate Targets panel:



```
 ADBPCMT n --------------- CM - Associate Targets ------------- Row 1 to 1 of 1

Details for multi-target change: VNDR1.S28479-C1        DB2 System: 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
    *        *             *         *                      *
--- -------> ------------> --------- ---------------------- -------------------
 I  TESTCHG1 DD1A          ATCOWN    ATCNAM                 NEW
******************************* END OF DB2 DATA *******************************
```

Figure 383. CM - Associate Targets Panel (ADBPCMT)

   Note: If no targets exist, panel ADBPC911 displays to allow you to insert targets.

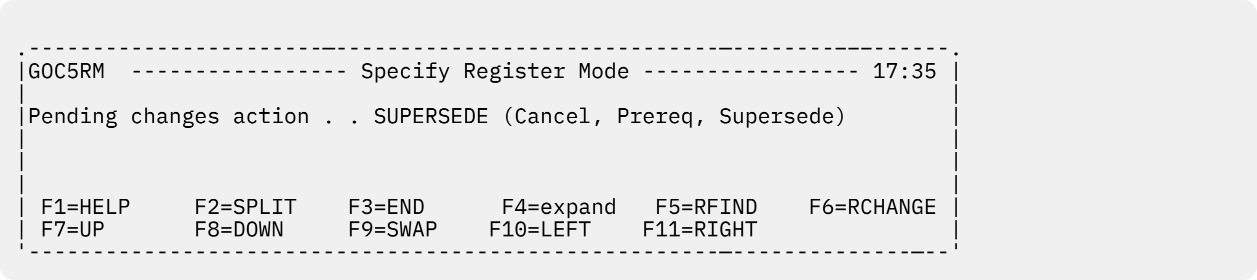
3. Specify the action to take for any pending changes to the objects on the target system that are affected by this change:

   Cancel

      Do not register the change if there are pending changes.

Prereq
Make the pending changes for the affected objects prerequisite changes for this change.

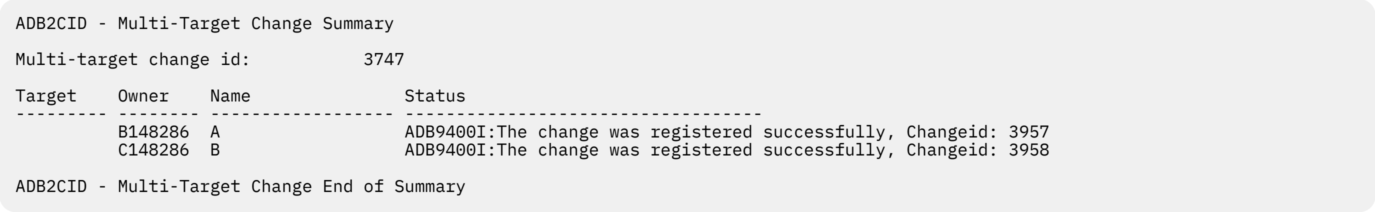
Supersede
Make this change a prerequisite change for the pending changes.



```
.-----------------------------------------------------------------------.
|GOC5RM  ----------------- Specify Register Mode ----------------- 17:35 |
|                                                                        |
|Pending changes action . . SUPERSEDE (Cancel, Prereq, Supersede)        |
|                                                                        |
|                                                                        |
|                                                                        |
| F1=HELP     F2=SPLIT    F3=END      F4=expand   F5=RFIND    F6=RCHANGE |
| F7=UP       F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT               |
'------------------------------------------------------------------------'
```

Figure 384. Specify Register Mode Panel (GOC5RM)

If successful, the output indicates **Register Successful** and the changes are registered on the specified targets. The following is an example report of a multi-target change summary:



```
ADB2CID - Multi-Target Change Summary

Multi-target change id:          3747

Target    Owner    Name               Status
--------- -------- ------------------ -----------------------------------
          B148286  A                  ADB9400I:The change was registered successfully, Changeid: 3957
          C148286  B                  ADB9400I:The change was registered successfully, Changeid: 3958

ADB2CID - Multi-Target Change End of Summary
```

Figure 385. Multi-Target Change (ADB2CID)

Related tasks

"Customizing Db2 Admin Tool" on page 91
After Db2 Admin Tool is installed, you can customize the configuration by running IBM Tools Customizer for z/OS (TCz).

Importing multi-target changes

You can view all of the change statements in a target file before they are imported to a target system. A *target file* contains only one change, but the change can have multiple statements.

About this task

Importing multi-target changes is similar to importing changes to a single target. An additional panel is displayed when importing multi-target changes.

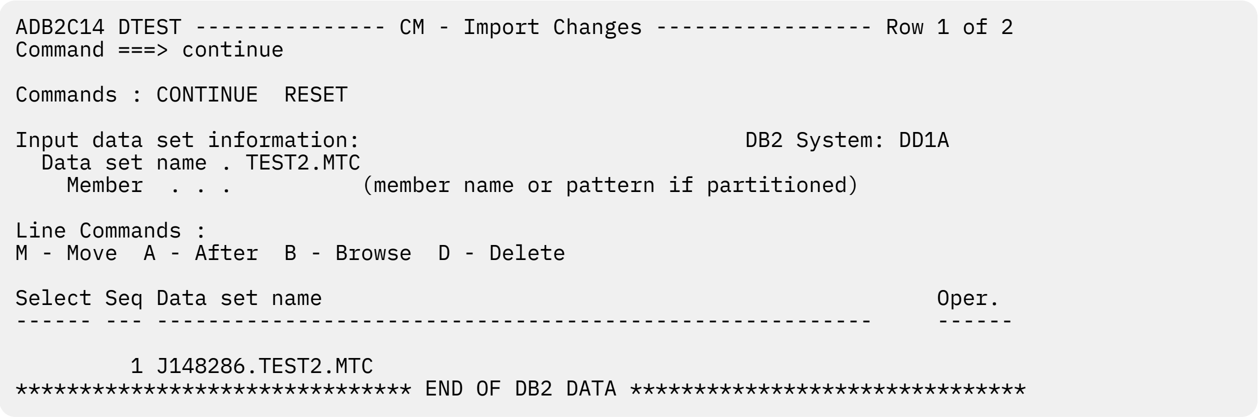
When you import changes to the local target, you can import the change statements that are contained in the file to a (single) local target Db2 subsystem.

To import a multi-target change:

Procedure

1. Specify option 1 on the **Change Management** panel to display the **Manage Changes** panel.
2. Specify option 4 to import changes.
3. Specify the name of the data set that contains the multi-target change statements (see "Multi-target changes" on page 823). Only a single data set containing multi-target change content can be imported at one time.

   The following figure shows the Import Changes panel:

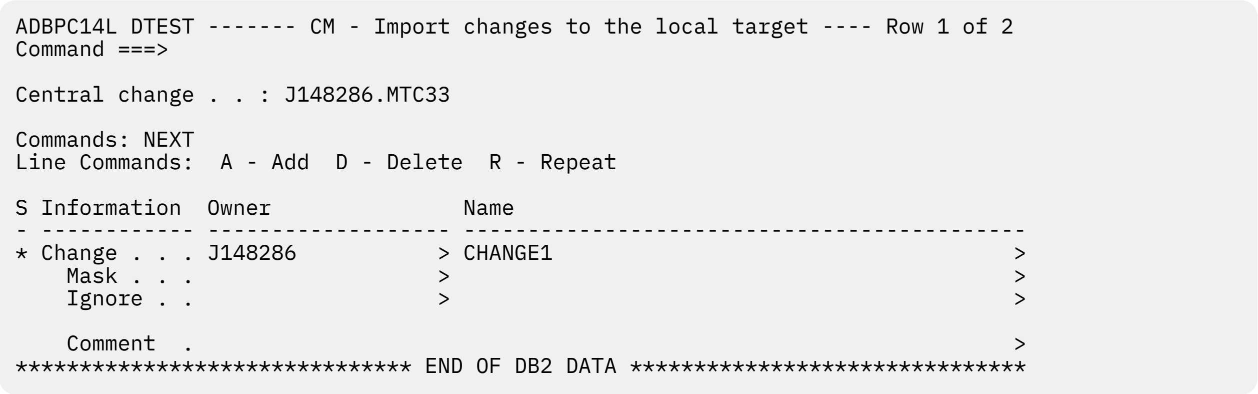


```
ADB2C14 DTEST --------------- CM - Import Changes ----------------- Row 1 of 2
Command ===> continue

Commands : CONTINUE  RESET

Input data set information:                              DB2 System: DD1A
  Data set name . TEST2.MTC
    Member  . . .          (member name or pattern if partitioned)

Line Commands :
M - Move  A - After  B - Browse  D - Delete

Select Seq Data set name                                                Oper.
------ --- --------------------------------------------------------     ------

         1 J148286.TEST2.MTC
******************************* END OF DB2 DATA *******************************
```

Figure 386. Import Changes panel (ADB2C14)

To process the import, issue the CONTINUE command. To clear the list of data sets, issue RESET.

4. If you are importing a multi-target change the **Import changes to the local target** panel is displayed, as shown in the following figure:

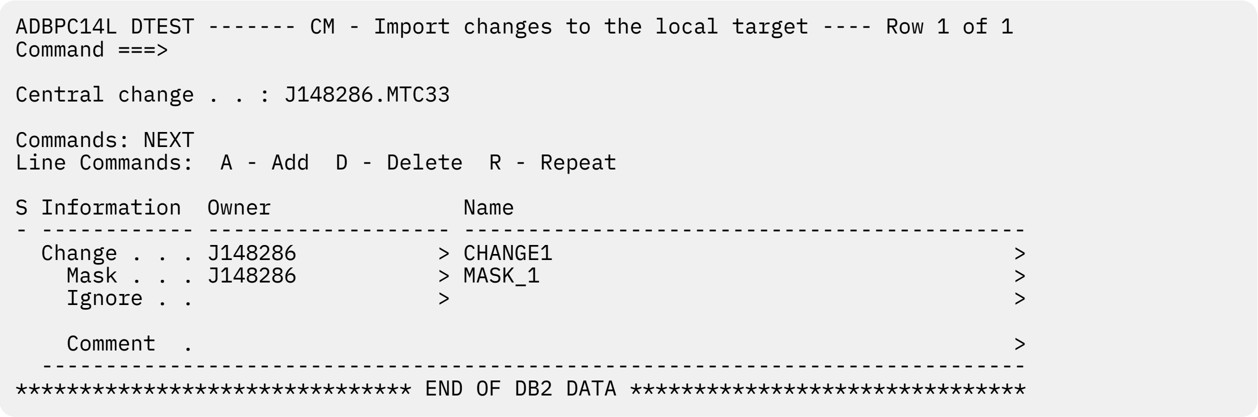


```
ADBPC14L DTEST ------- CM - Import changes to the local target ---- Row 1 of 2
Command ===>

Central change . . : J148286.MTC33

Commands: NEXT
Line Commands:  A - Add  D - Delete  R - Repeat

S Information  Owner               Name
- ------------ ------------------- --------------------------------------------
* Change . . . J148286           > CHANGE1                                    >
    Mask . . .                   >                                            >
    Ignore . .                   >                                            >

    Comment  .                                                                >
******************************* END OF DB2 DATA *******************************
```

Figure 387. Import Changes to the local target panel (ADBPC14L)

The target change fields are input fields, allowing you to override the contents of the multi-target change file.

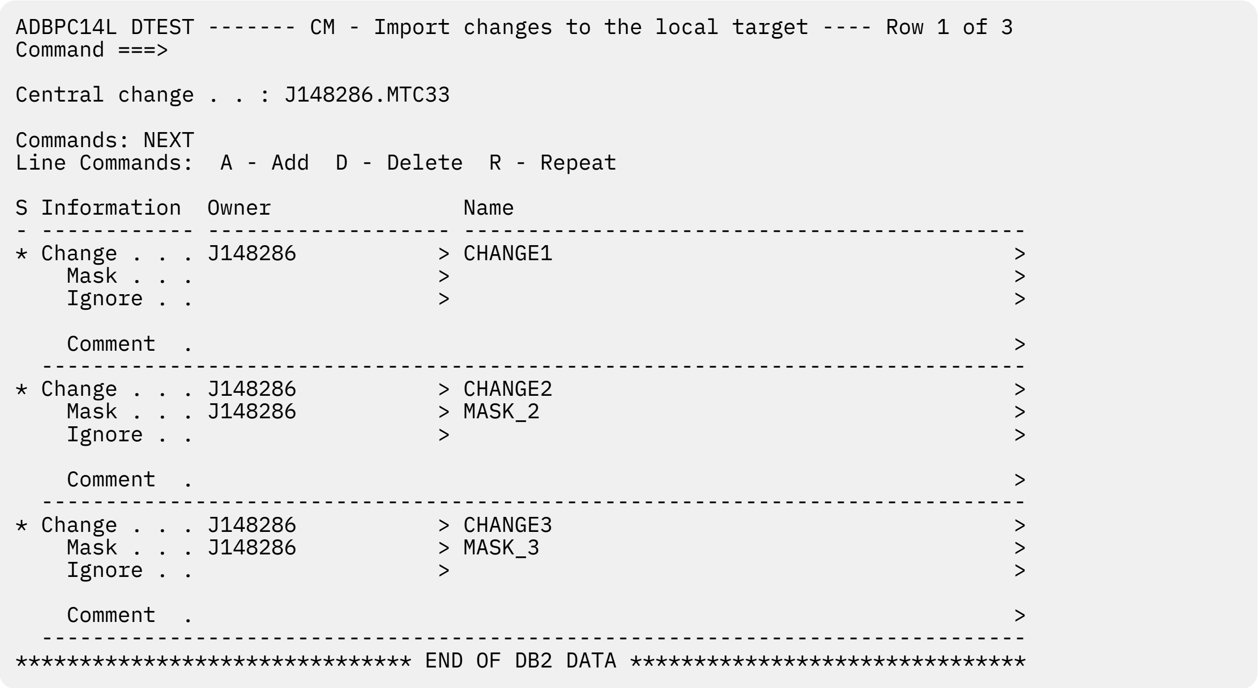
5. You can use masking to affect different objects. A mask allows you to change the object names as they are read from the file, which allows you to affect a different set of objects on the target. You specify the mask name and owner, as in the following figure. Note that optional ignores can also be specified while registering the change; however, ignores will be applied while analyzing the change.



```
ADBPC14L DTEST ------- CM - Import changes to the local target ---- Row 1 of 1
Command ===>

Central change . . : J148286.MTC33

Commands: NEXT
Line Commands:  A - Add  D - Delete  R - Repeat

S Information  Owner               Name
- ------------ ------------------- --------------------------------------------
  Change . . . J148286           > CHANGE1                                    >
    Mask . . . J148286           > MASK_1                                     >
    Ignore . .                   >                                            >

    Comment  .                                                                >
  -----------------------------------------------------------------------------
******************************* END OF DB2 DATA *******************************
```

Figure 388. Import Changes to the local target panel (ADBPC14L)

6. Use the A (Add) or R (Repeat) line commands to specify additional changes. For example, two additional changes have been added in the following figure:



```
ADBPC14L DTEST ------- CM - Import changes to the local target ---- Row 1 of 3
Command ===>

Central change . . : J148286.MTC33

Commands: NEXT
Line Commands:  A - Add  D - Delete  R - Repeat

S Information  Owner               Name
- ------------ ------------------- ----------------------------------------------
* Change . . . J148286           > CHANGE1                                      >
    Mask . . .                   >                                              >
    Ignore . .                   >                                              >

    Comment  .                                                                  >
  ------------------------------------------------------------------------------
* Change . . . J148286           > CHANGE2                                      >
    Mask . . . J148286           > MASK_2                                       >
    Ignore . .                   >                                              >

    Comment  .                                                                  >
  ------------------------------------------------------------------------------
* Change . . . J148286           > CHANGE3                                      >
    Mask . . . J148286           > MASK_3                                       >
    Ignore . .                   >                                              >

    Comment  .                                                                  >
  ------------------------------------------------------------------------------
******************************* END OF DB2 DATA *******************************
```

Figure 389. Import Changes to the local target panel (ADBPC14L)

7. Issue the NEXT command.

   When entered, the NEXT command builds a batch job that registers the change(s) on the system. After submitting the batch job you can display your imported change on the **Changes** panel, analyze the change, and then run it.

   Each change owner and name specified on the panel must be unique because the change statements in the file are imported to the same Db2 subsystem. Even if you use unique change owner/names, you should not have the same objects affected by the same change statements more than one time. You should specify different masks to affect changes to different objects.

   You can determine whether a change has already been registered with the same multi-target change ID or the same mask as the one you are importing. If an existing change is identified, it will be identified with one of the following statuses:

   Initial
   The change will be restarted. Supersede and prerequisite decisions will be used.

   Defined, Analyzed, Complete, or Running:
   The change will remain as it is currently defined. The input change is ignored.

   Canceled
   The input change is registered.

   The input change name and owner should not conflict with the existing canceled change.

Exporting multi-target information to a data set on the target system

When a target system does not have DRDA connectivity to the central system, the continuous updates for the multi-target changes are not communicated to the central system. You can consolidate updates for target changes into a data set. This data set can eventually be processed on the central system so that the central system is synchronized with the target systems.

About this task

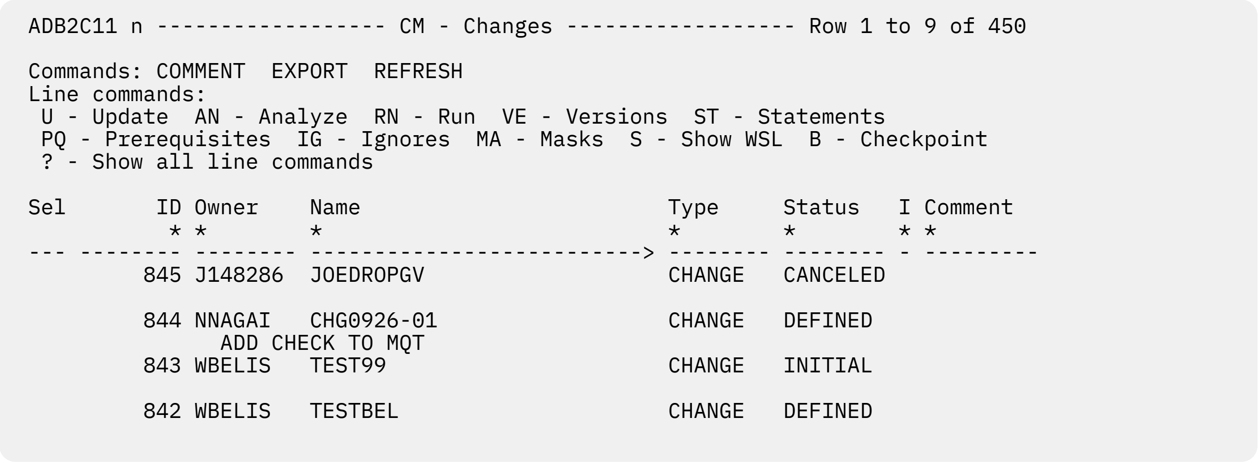
You can consolidate status updates for parameters, selection criteria, and other related options.

To export a multi-target change, you can use either a batch process or online process.

Procedure

To export multi-target information to a data set on the target system:

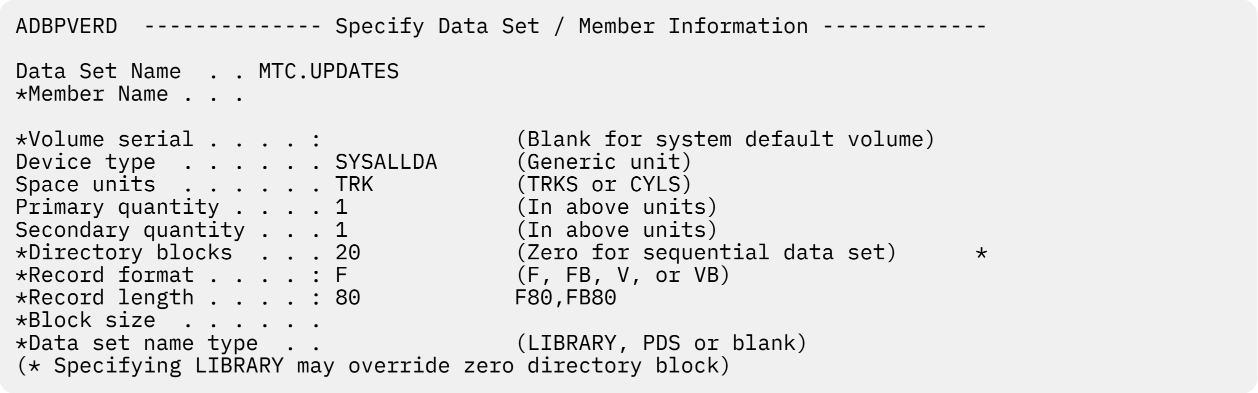
1. On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option 1, and press Enter.
3. On the **Manage Changes (ADB2C1)** panel, specify one of the following options:
   - To use a batch process to export the multi-target change, specify option 6.
   - To use an online process to export the multi-target change, specify option 1.
4. If you are using the online process, on the **CM - Changes (ADB2C11)** panel, specify the EXPORT command.



```
ADB2C11 n ------------------ CM - Changes ------------------ Row 1 to 9 of 450

Commands: COMMENT  EXPORT  REFRESH
Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks  S - Show WSL  B - Checkpoint
 ? - Show all line commands

Sel       ID Owner    Name                        Type     Status   I Comment
           * *        *                           *        *        * *
--- -------- -------- ------------------------->  -------- -------- - ---------
         845 J148286  JOEDROPGV                   CHANGE   CANCELED

         844 NNAGAI   CHG0926-01                  CHANGE   DEFINED
               ADD CHECK TO MQT
         843 WBELIS   TEST99                      CHANGE   INITIAL

         842 WBELIS   TESTBEL                     CHANGE   DEFINED
```

*Figure 390. **CM - Changes (ADB2C11)** panel*

5. On the **Specify Data Set / Member Information (ADBPVERD)** panel, specify the name of the data set that is to contain the target information, and press Enter:



```
ADBPVERD  -------------- Specify Data Set / Member Information -------------

Data Set Name  . . MTC.UPDATES
*Member Name . . .

*Volume serial . . . . :               (Blank for system default volume)
Device type  . . . . . . SYSALLDA      (Generic unit)
Space units  . . . . . . TRK           (TRKS or CYLS)
Primary quantity . . . . 1             (In above units)
Secondary quantity . . . 1             (In above units)
*Directory blocks  . . . 20            (Zero for sequential data set)      *
*Record format . . . . : F             (F, FB, V, or VB)
*Record length . . . . : 80            F80,FB80
*Block size  . . . . . .
*Data set name type  . .               (LIBRARY, PDS or blank)
(* Specifying LIBRARY may override zero directory block)
```

6. If the data set already exists, on the **Replace Data Set Confirmation (ADB2CONF)** panel, specify whether you want to replace the contents of the data set or cancel, and press Enter.
7. In the generated job, change any of the following SYSIN parameters as needed:

   ALTERAGE

   Specifies a time period to filter the target changes. Only those target changes that were altered during the specified period are exported. Acceptable formats are: *n* YEAR(S), *n* MONTH(S), *n* DAY(S), *n* MINUTE(S), or *n* SECOND(S).

   Example: `ALTERAGE="1 MONTH"`

   MTCLOCS

   Specifies multi-target central locations. Use a comma to separate each value.

   Example: `MTCLOCS="'DSNA','DSNB','DSNC'"`

MTCIDS
Specifies multi-target change ID values. Use a comma to separate each value.

Example: `MTCIDS="10, 1000, 3100"`

CHGIDS
Specifies target change ID values. Use a comma to separate each value.

Example: `CHGIDS="1, 11, 40, 1001"`

8. Run the job.

   The changes are exported to the specified data set.

   After the job runs successfully, the output file contains multi-target information, as shown in the following example:

```
VIEW       VIJAYAK.MTC.UPDATES                             Columns 00001 00072
Command ===>                                                  Scroll ===>; CSR
****** ***************************** Top of Data ******************************
000001 <TARGETINFO VERSION="1">
000002 <MTCLOCATION>DSNA
000003 <TARGETLOCATION>DSNB
000004 <CHANGE>
000005 <MTCCHANGEID>2578</MTCCHANGEID>
000006 <OWNER>QMFADM</OWNER>
000007 <NAME>AUTO:2013-06-14-09.07.46.578784</NAME>
000008 <STATUS>INITIAL</STATUS>
000009 <MASK>
000010 <OWNER></OWNER>
000011 <NAME></NAME>
000012 </MASK>
000013 <CREATEDTS>2013-06-14-09.07.47.824553</CREATEDTS>
000014 <ALTEREDTS>2013-06-14-09.07.47.824553</ALTEREDTS>
000015 </CHANGE>
000016 <CHANGE>
000017 <MTCCHANGEID>2637</MTCCHANGEID>
000018 <OWNER>QMFADM</OWNER>
000019 <NAME>AUTO:2013-06-15-17.41.44.561870</NAME>
```

   A report is also generated, as shown in the following example:

```
********************************* TOP OF DATA **********************************

Multi-target changes Report:
============================

< MTC Details:    >  < Target Details:                                Altered
Location   ChangeID  ChangeID   Owner    Change Name        Status    Timestamp
--------  ---------- ---------- -------- ------------------ --------- ----------
DSNA            2578        226 QMFADM   AUTO:2013-06-14-09 INITIAL   2013-06-14

DSNA            2637        280 QMFADM   AUTO:2013-06-15-17 DEFINED   2013-06-15

DSNA            2674        292 QMFADM   AUTO:2013-06-18-14 DEFINED   2013-06-18

DSNB             259        260 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB             261        262 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB             263        264 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB             265        266 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB             267        268 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB             269        270 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB             271        272 VIJAYAK  AUTO:2013-06-15-16 INITIAL   2013-06-15

DSNB             274        275 VIJAYAK  AUTO:2013-06-15-17 DEFINED   2013-06-15

DSNB             277        278 VIJAYAK  AUTO:2013-06-15-17 DEFINED   2013-06-15

DSNB             285        286 VIJAYAK  AUTO:2013-06-15-18 ANALYZED  2013-06-15

DSNB             287        288 J148286  AUTO:2013-06-17-10 DEFINED   2013-06-18

******************************** BOTTOM OF DATA
********************************
```

If the search criteria resulted in no rows found, a warning is displayed, and the job ends with RC=8.

Importing multi-target information from a data set on the central system

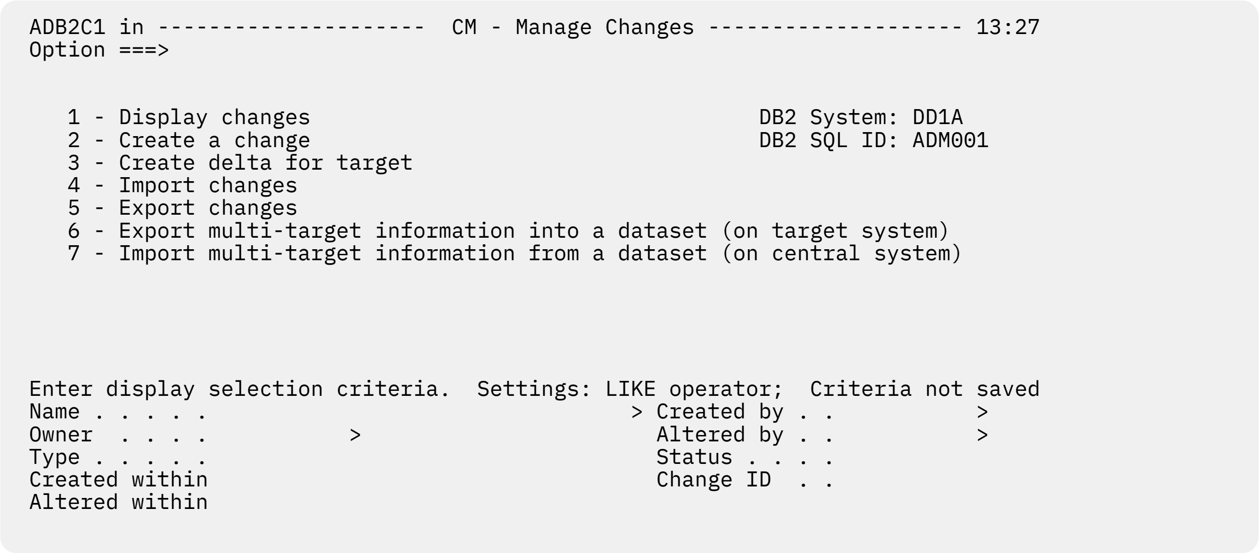
You can use a batch interface to process a status update file on a central multi-target system so that the central system will be synchronized with the target systems for the targets that are associated with the central system.

About this task

To import a multi-target change information from a data set:

Procedure

1. Specify option 1 on the **Change Management** panel to display the **Manage Changes** panel as shown in the following figure:

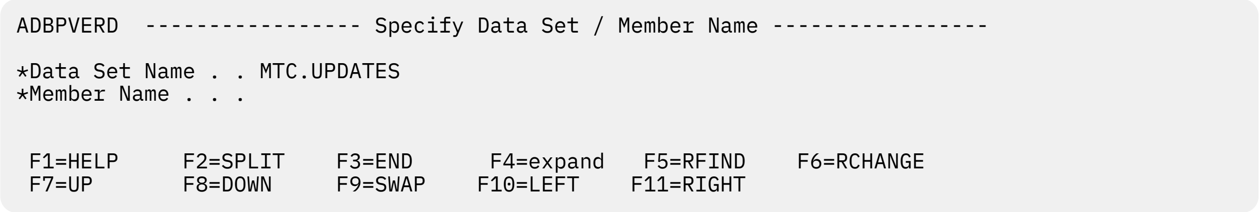


```
ADB2C1 in ---------------------  CM - Manage Changes -------------------- 13:27
Option ===>


   1 - Display changes                                  DB2 System: DD1A
   2 - Create a change                                  DB2 SQL ID: ADM001
   3 - Create delta for target
   4 - Import changes
   5 - Export changes
   6 - Export multi-target information into a dataset (on target system)
   7 - Import multi-target information from a dataset (on central system)





Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . .                           > Created by . .           >
Owner  . . . .           >                 Altered by . .           >
Type . . . . .                             Status . . . .
Created within                             Change ID  . .
Altered within
```

Figure 391. Manage Changes panel (ADB2C1)

2. Specify option 7, Import multi-target information from a dataset (on central system).

   The Specify Data Set / Member Name panel (ADBPVERD) is displayed as shown in the following example:



```
ADBPVERD  ----------------- Specify Data Set / Member Name -----------------

*Data Set Name . . MTC.UPDATES
*Member Name . . .


 F1=HELP     F2=SPLIT    F3=END      F4=expand   F5=RFIND    F6=RCHANGE
 F7=UP       F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT
```

Figure 392. Specify Data Set / Member Name panel (ADBPVERD)

3. Specify the dataset (and member) where the target information is to be exported.

   A new job is created. When the job is submitted, should update the target information on central multi-target system. The appropriate entries in the ADBCHGAT table will be updated.

   A report is also generated by the job (dd REPORT) The purpose of the report is to allow you to determine what action should be taken for each entry in the file. You can specify `REPORT_LEVEL=All`, `Current Location`, or `Updated`.

   All
   All entries pertaining to all MTC locations are displayed.

   Current® Location
   All entries pertaining to the current location are displayed.

   Updated
   Only updated entries are displayed. The entries are a subset of the entries for the current location. This is the default setting.

   If a search criteria results in no rows found, a warning is displayed and the process ends with RC=8.

Recovering changes that are made through Change Management

A change is recoverable if a corresponding recover change exists. A *recover change* is a change that is generated by Db2 Admin Tool to back out a completed change. You can request that a recover change be generated when a change analyzed.

Before you begin

To recover a change, all of the following criteria must be true:

- The change must be in COMPLETE status.
- A recover change must exist for the change and be in ANALYZED status. The WSL that was generated for the recover change during the analyze process must also be available.
- All completed changes that must be recovered first have been recovered. For example, assume that you made the following changes:

  1. Created a table space.
  2. Created a table in the table space.
  3. Modified the table to insert a new column.

  If you want to recover the change that created the table space (which means dropping the table space), you must first recover the change to insert the new column into the table and then recover the change to create the table. Each of these changes must have a corresponding recover change.

Restriction: The following restrictions apply to recovering changes:

- If an ignore was specified for a change, the change cannot be recovered.
- If privileges were granted as part of the change to be recovered, the privileges are not revoked when the change is recovered. You must create a new change to revoke the privileges. Changes to revoke privileges can be made through Change Management only if they are run as immediate changes.
- If you rotate a table partition multiple times, you can only recover the most recent change.

About this task

Changes must be backed out one at a time. When you attempt to recover a change, Db2 Admin Tool identifies any completed changes that must be recovered first and lists them in the order in which you need to recover them. The list of changes represents those changes that completed after the change to be recovered completed and that modify the same or a related set of objects as the change to be recovered.

Procedure

To recover a change:

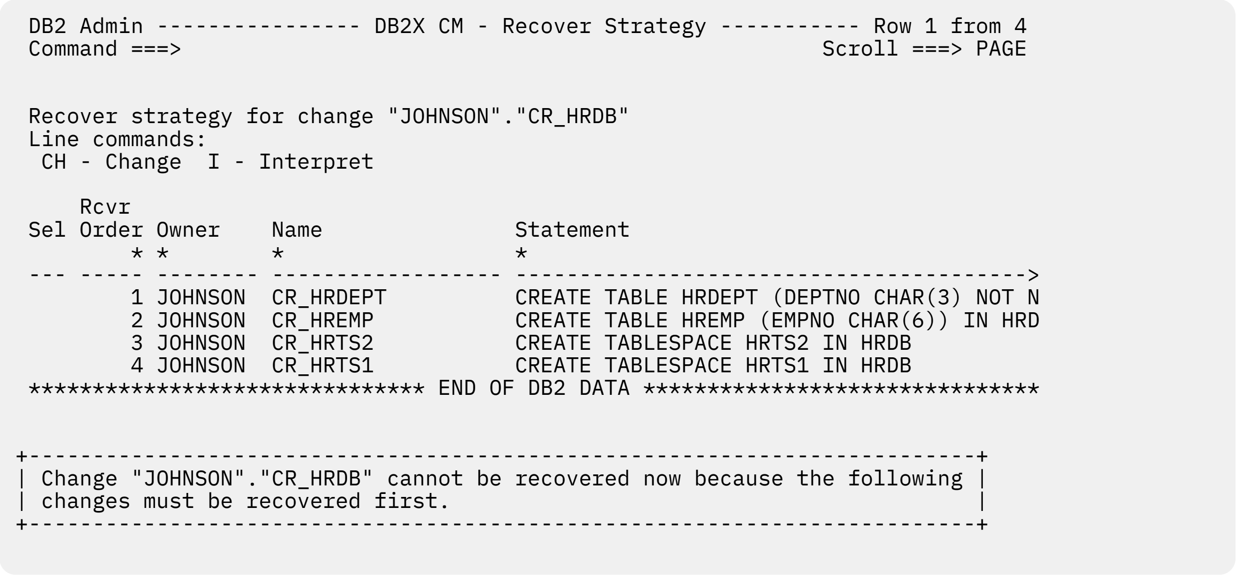
1. Navigate to the Change Management panels:

   On the **DB2 Administration Menu (ADB2)** panel, specify option `CM`, and press Enter.
2. Display a list of changes:
   a) On the **Change Management (CM) (ADB2C)** panel, specify option 1, and press Enter.
   b) On the **Manage Changes (ADB2C1)** panel, specify option 1, and press Enter
3. On the **Changes (ADB2C11)** panel, issue the RC line command for the change that you want to recover.

   Important: You cannot issue the run line command (RN) for the recover change itself. Instead, you always recover a change by issuing the recover line command (RC) for the change to recover.

   Db2 Admin Tool prompts you in the following situations:

   - If the change has a recover change for both the original and existing data, you are prompted to choose which data you want to use for the recovery.
   - If the change cannot be recovered because it has no recover change (or the change that must be recovered first does not have a recover change), an error message is issued.
   - If the change cannot be recovered because other changes must be recovered first, a panel is displayed with the list of changes that must be recovered first and the order in which the changes must be recovered, as shown in the following example:

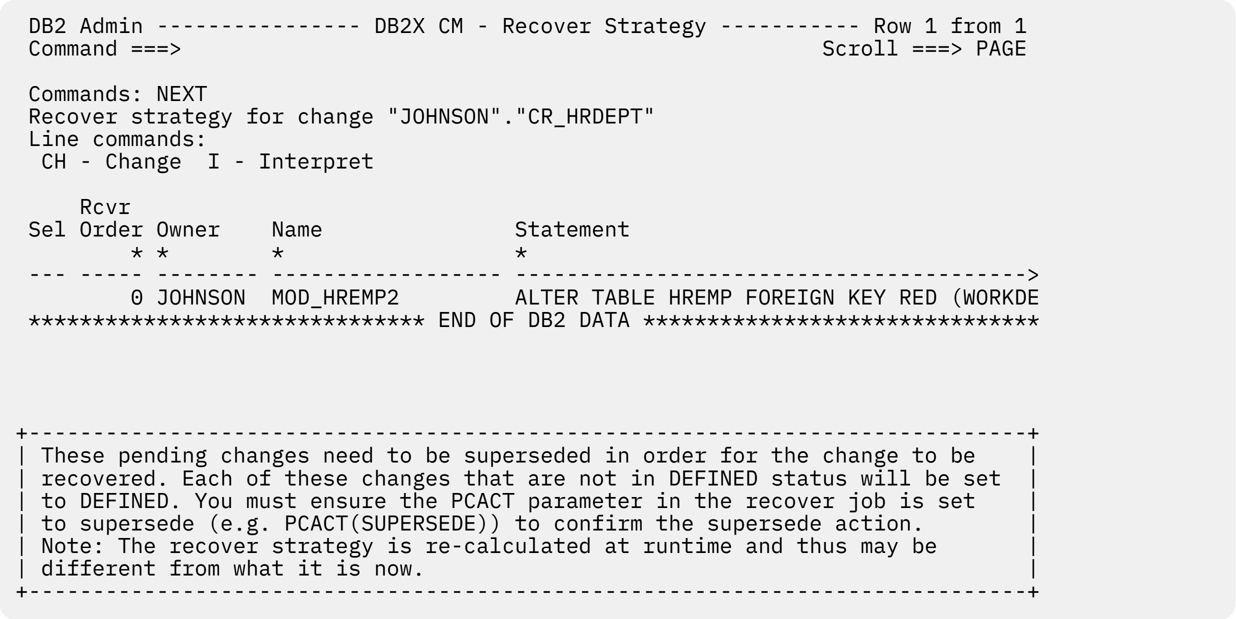


```
DB2 Admin ---------------- DB2X CM - Recover Strategy ----------- Row 1 from 4
Command ===>                                                  Scroll ===> PAGE


Recover strategy for change "JOHNSON"."CR_HRDB"
Line commands:
 CH - Change  I - Interpret

    Rcvr
Sel Order Owner    Name               Statement
        * *        *                  *
--- ----- -------- ------------------ --------------------------------------->
        1 JOHNSON  CR_HRDEPT          CREATE TABLE HRDEPT (DEPTNO CHAR(3) NOT N
        2 JOHNSON  CR_HREMP           CREATE TABLE HREMP (EMPNO CHAR(6)) IN HRD
        3 JOHNSON  CR_HRTS2           CREATE TABLESPACE HRTS2 IN HRDB
        4 JOHNSON  CR_HRTS1           CREATE TABLESPACE HRTS1 IN HRDB
******************************* END OF DB2 DATA *******************************


+-----------------------------------------------------------------------+
| Change "JOHNSON"."CR_HRDB" cannot be recovered now because the following |
| changes must be recovered first.                                      |
+-----------------------------------------------------------------------+
```

Figure 393. Example of list of changes that must be recovered

Recover the list of changes in the order that is specified before you recover this change.

- If the change can be recovered but recovering the change will cause other changes in ANALYZED status to be set to DEFINED status, a panel is displayed with the list of changes that will be set to DEFINED status, as shown in the following example:



```
DB2 Admin ---------------- DB2X CM - Recover Strategy ----------- Row 1 from 1
Command ===>                                                  Scroll ===> PAGE

Commands: NEXT
Recover strategy for change "JOHNSON"."CR_HRDEPT"
Line commands:
 CH - Change  I - Interpret

    Rcvr
Sel Order Owner    Name               Statement
        * *        *                  *
--- ----- -------- ------------------ --------------------------------------->
        0 JOHNSON  MOD_HREMP2         ALTER TABLE HREMP FOREIGN KEY RED (WORKDE
******************************* END OF DB2 DATA *******************************




+---------------------------------------------------------------------------+
| These pending changes need to be superseded in order for the change to be    |
| recovered. Each of these changes that are not in DEFINED status will be set  |
| to DEFINED. You must ensure the PCACT parameter in the recover job is set    |
| to supersede (e.g. PCACT(SUPERSEDE)) to confirm the supersede action.        |
| Note: The recover strategy is re-calculated at runtime and thus may be       |
| different from what it is now.                                               |
+---------------------------------------------------------------------------+
```

Figure 394. Example of list of changes that will be set to DEFINED status

If this panel is displayed, review the list of changes. Then, issue the NEXT command to proceed with recovering the change.

4. Edit and submit the generated job.

   When the job completes successfully, the status of the change that is recovered is set to DEFINED and the status of the recover change is set to COMPLETE.

   If the job fails, check the job output to determine the cause of failure, make the necessary corrections, and restart the job.

5. Press PF3 to return to the **Changes (ADB2C11)** panel to verify that the status of the change is DEFINED and the status of the recover change is COMPLETE.

Tip: If you return to the **Changes (ADB2C11)** panel before the submitted job completes, you can click the REFRESH command after the job completes to see the refreshed status of the change.

Related tasks

"Analyzing a change" on page 652
After a change is registered, you must analyze it before you can run it. During this analyze step, Db2 Admin Tool analyzes how the change modifies existing objects and creates a work statement list (WSL) that can be used to run the change.

Modifying changes

You can modify the change statements in an existing change if the change is in INITIAL, DEFINED, or ANALYZED status.

About this task

However, modifying an existing change is considered a manual intervention and is not recommended for several reasons. When you modify an existing change, Db2 Admin Tool cannot apply virtual changes or determine whether pending changes exist. Modifying an existing change can also impact other existing changes substantially. For example, the change you are modifying might be a pending change that was applied when another change was created.

During the process of modifying a change, Db2 Admin Tool checks only the syntax of each change statements. When you modify change statements through the Change Statements panel (ADB2C1S) panel, for example, syntax checking is completed at the time that you exit the panel. Semantic checking is done during the analyze process.

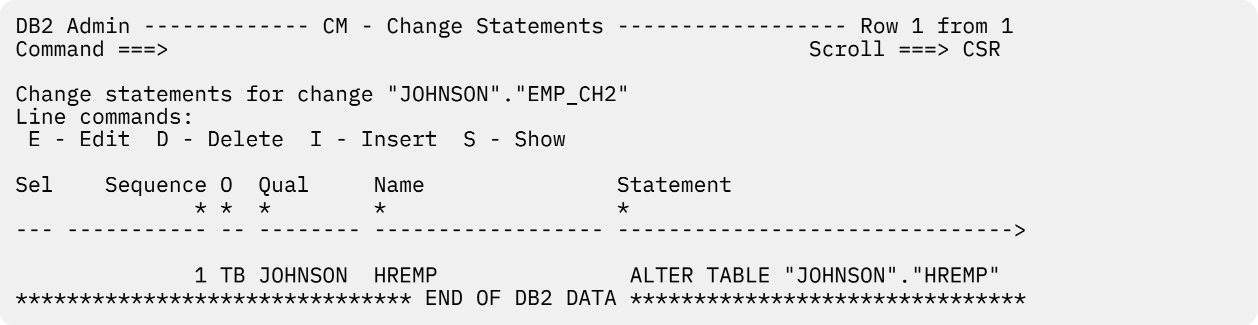
To modify the change statements in an existing change:

Procedure

1. Identify and consider the impact of the changes to dependent changes.

   For example, assume that want to modify a change that adds a new column to a table to change the name of the column. The change might be a prerequisite change to other changes that use that column such as another change that creates an index that includes that column.
2. Display the change to be modified by selecting option 1 on the **Change Management** panel, and then select option 1 on the **Manage Changes** panel.
3. Issue the ST line command to display the change statements in the change that you want to modify on the **Change Statements** panel.

   The following figure shows an example of the **Change Statements** panel:



```
DB2 Admin ------------- CM - Change Statements ------------------ Row 1 from 1
Command ===>                                                  Scroll ===> CSR

Change statements for change "JOHNSON"."EMP_CH2"
Line commands:
 E - Edit  D - Delete  I - Insert  S - Show

Sel    Sequence O  Qual     Name               Statement
              * *  *        *                  *
--- ----------- -- -------- ------------------ ------------------------------->

              1 TB JOHNSON  HREMP               ALTER TABLE "JOHNSON"."HREMP"
****************************** END OF DB2 DATA ******************************
```

Figure 395. Change Statements panel (ADB2C1S)

4. Issue the E line command to change any of change statements in the change, the D line command to delete a change statement, and the I line command to insert a new change statement.

   When you use the E and I line commands, you are put into an ISPF edit session and can work with the SQL statement.
5. Press F3 to return to the **Change Statements** panel.

Db2 Admin Tool reregisters the change. A message is displayed to indicate whether the change was registered successfully. When a modified change is reregistered, pending changes or prerequisite changes are not processed.

6. Reanalyze any change that is in ANALYZED status and that is impacted by the modifications that you made to this change.

   Reanalyzing the impacted changes ensures the validity of the changes.

Deleting changes

You can delete certain types of changes if Db2 Admin Tool has been configured to support the delete change line command and you have the appropriate privileges.

About this task

If the requested change is deleted and has a recover change, the recover change is also deleted.

You can delete only changes that have a type of COMPARE, FAST, CHANGE, or RECOVER.

To delete a change:

Procedure

1. Display the change to be deleted by selecting option 1 on the **Change Management** panel, and then select option 1 on the **Manage Changes** panel.
2. Issue the DEL line command against the change that you want to delete.

   A pop-up window is displayed to confirm your intention to delete the change.
3. Select 2 to continue with deleting the change.

Results

After a change is deleted, change no longer appears in the list of changes. The change is removed from the Change Management database, which removes any audit tracking for the change.

Promoting changes

Promoting changes allows you to move changes from one system to another because a delta changes data set is generated, which you can then import into a change on another system.

About this task

To promote a change, two versions must exist. The *starting version* represents the state of objects before any changes are made and the *ending version* represents the state of objects after the promoted changes are made. During the promote process, Db2 Admin Tool compares the ending version with the starting version and generates a delta changes data set the contains the SQL statements that are required to bring the other system up to the same level as the system from which your promoting the changes.

To promote a change:

Procedure

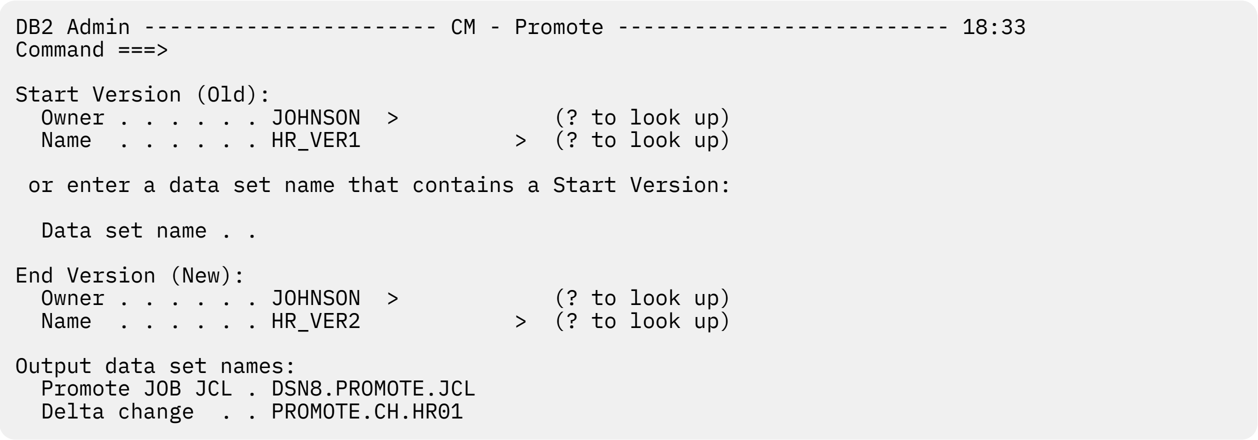
1. Specify option 1 on the **Change Management** panel, and then select option 3 on the **Manage Changes** panel to display the **Promote** panel.

   Alternatively, you can use either of the following methods to display the **Promote** panel:

   - If you know the ending version, specify the PR line command for the version on the **Versions** panel. The **Promote** panel will be displayed with the information for the ending version filled in.
   - If you know the change and a new base version was created when the change was run, specify the PR line command for the change on the **Changes** panel. The **Promote** panel will be displayed with the information for the ending version filled in.

2. Specify the following information on the **Promote** panel and press Enter.
   - The starting version
   - The ending version
   - The data set name for the promote batch job
   - The data set name for the delta changes statements

   The following figure shows an example of the **Promote** panel:



```
DB2 Admin ---------------------- CM - Promote ------------------------- 18:33
Command ===>

Start Version (Old):
  Owner . . . . . . JOHNSON  >            (? to look up)
  Name  . . . . . . HR_VER1            >  (? to look up)

 or enter a data set name that contains a Start Version:

  Data set name . .

End Version (New):
  Owner . . . . . . JOHNSON  >            (? to look up)
  Name  . . . . . . HR_VER2            >  (? to look up)

Output data set names:
  Promote JOB JCL . DSN8.PROMOTE.JCL
  Delta change  . . PROMOTE.CH.HR01
```

   Figure 396. Promote (ADB2CPS) panel

3. Specify the following information on the register panel and issue the CONTINUE command:
   a. Specify an owner and a name for the change. The default owner is the current SQL ID.
   b. Optionally, specify a comment for the change, an ignore for the change, and a mask for the change.

   The change will be registered as a COMPARE change.
4. Edit and submit the generated job.

   When the job completes successfully, the change is placed in COMPLETE status.

Results

You can now import the delta changes data set into a new change on another system, analyze the change, and run the change to bring the level of the other system up to the level of the current system.

Exporting changes

You can selectively export multiple changes made in one environment so that you can later distribute those changes to multiple external environments.

About this task

To export changes, you must first create a list of the changes whose statements are to be promoted. You can arrange those changes in any desired sequence. When the list is complete and you request that those changes be exported, Db2 Admin Tool extracts all of the change statements to a single file. A new change is created with the type COMPARE; it is marked COMPLETE when the promote process is complete. That file with the change statements can be then be imported into different environment. The import function uses the statements in that file to implement the change in the target environment.

You can create a single change by exporting multiple files at the same time. All types can be part of the same export.

Exporting a change is a two-phase process in which Db2 Admin Tool determines if the objects have any pending changes and then registers the exported change. If the exported SQL statements affect objects that have pending changes, the system determines whether the change becomes a prerequisite change for those pending changes.

Requirement: When you export SQL statements into a change, the version of Db2 that is on the system must support the SQL statements that you are exporting.

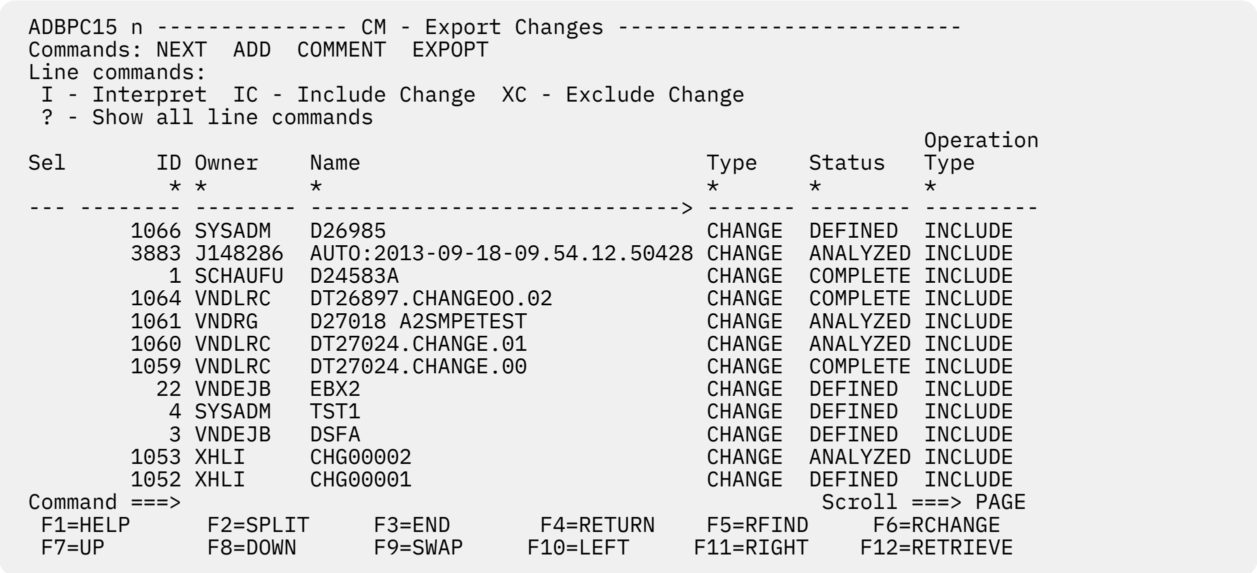
Export processing can be done in TSO or batch mode. TSO is the default.

Procedure

To export a change:

1. Navigate to the **CM - Export Changes (ADBPC15)** panel by completing one of the following sets of steps:
   - **If you want to export multiple changes:**
     a. On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
     b. On the **Change Management (CM) (ADB2C)** panel, specify option 1, and press Enter.
     c. On the **Manage Changes (ADB2C1)** panel, specify option 5, and press Enter.
   - **If you want to export a single change:**
     a. On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
     b. On the **Change Management (CM) (ADB2C)** panel, specify option 1, and press Enter.
     c. On the **Manage Changes (ADB2C1)** panel, specify option 1 and optionally any filtering criteria at the bottom, and press Enter.
     d. On the **CM - Changes (ADB2C11)** panel, issue the EX line command next to the change that you want to export.
2. On the **CM - Export Changes (ADBPC15)** panel, issue the XC line command to exclude individual changes from the list as needed.

   If you navigated to this panel by selecting option 5 on the **Manage Changes (ADB2C1)** panel, all changes are listed and are marked as INCLUDE by default.



```
ADBPC15 n --------------- CM - Export Changes ---------------------------
Commands: NEXT  ADD  COMMENT  EXPOPT
Line commands:
 I - Interpret  IC - Include Change  XC - Exclude Change
 ? - Show all line commands
                                                                      Operation
Sel       ID Owner    Name                           Type    Status   Type
           * *        *                              *       *        *
--- -------- -------- -----------------------------> ------- -------- ---------
        1066 SYSADM   D26985                         CHANGE  DEFINED  INCLUDE
        3883 J148286  AUTO:2013-09-18-09.54.12.50428 CHANGE  ANALYZED INCLUDE
           1 SCHAUFU  D24583A                        CHANGE  COMPLETE INCLUDE
        1064 VNDLRC   DT26897.CHANGEOO.02            CHANGE  COMPLETE INCLUDE
        1061 VNDRG    D27018 A2SMPETEST              CHANGE  ANALYZED INCLUDE
        1060 VNDLRC   DT27024.CHANGE.01              CHANGE  ANALYZED INCLUDE
        1059 VNDLRC   DT27024.CHANGE.00              CHANGE  COMPLETE INCLUDE
          22 VNDEJB   EBX2                           CHANGE  DEFINED  INCLUDE
           4 SYSADM   TST1                           CHANGE  DEFINED  INCLUDE
           3 VNDEJB   DSFA                           CHANGE  DEFINED  INCLUDE
        1053 XHLI     CHG00002                       CHANGE  ANALYZED INCLUDE
        1052 XHLI     CHG00001                       CHANGE  DEFINED  INCLUDE
Command ===>                                                  Scroll ===> PAGE
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 397. ***CM - Export Changes (ADBPC15)*** *panel*

3. Optional: To specify a mask or change whether batch mode is used, complete the following steps:
   a) Issue the EXPOPT command, and press Enter.
   b) On the **Export Options (ADBPC15O)** panel, specify the relevant options, and press Enter:

```
ADBPC15O --------------------- DD1A Export Options -------------------- 10:52
Option ===>

Please specify the following Export options:

Export changes in batch . . . . . . . YES (Yes/No)

Enter mask details (optional):
  Mask Table Entry:
    Owner . . . .      >                  (? to look up)
    Name  . . . .                 >       (? to look up)
  Data Set:
    Mask DSN  . .
  Options:
    Edit Mask . .                         (Yes/No)

Show this panel prior to each use . . YES (Yes/No)
```

Figure 398. ***Export Options (ADBPC15O)*** *panel*

If you specify a mask, the mask is applied to the changes that you selected on the **CM - Export Changes (ADBPC15)** panel. The exported data set will have the specified mask applied.

Tip: If you want this options panel to be always be displayed after the **CM - Export Changes (ADBPC15)** panel, change the last field to YES.

4. On the **CM - Export Changes (ADBPC15)** panel, issue the NEXT command, and press Enter.

   If the **Export Options (ADBPC15O)** panel is displayed, specify any options on that panel as needed, and press Enter.
5. On the **Specify Data Set / Member Information (ADBPVERD)** panel, specify parameters for the data set that is to contain the final list of exported changes, and press Enter: (This data set is a changes file that can later be imported to a different environment.)

```
ADBPVERD  ------------- Specify Data Set / Member Information -------------

Data Set Name  . . EXPORTED.CHANGES
*Member Name . . .

*Volume serial . . . . :              (Blank for system default volume)
Device type  . . . . . . SYSALLDA     (Generic unit)
Space units  . . . . . . TRACKS       (TRKS or CYLS)
Primary quantity . . . . 1            (In above units)
Secondary quantity . . . 1            (In above units)
*Directory blocks  . . . 0            (Zero for sequential data set)      *
*Record format . . . . : F            (F or V)
*Record length . . . . : 80           F80
*Block size  . . . . . .
*Data set name type  . .              (LIBRARY, PDS or blank)
(* Specifying LIBRARY may override zero directory block)

 F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
 F7=UP       F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT
```

Figure 399. ***Specify Data Set / Member Information (ADBPVERD)*** *panel*

Exporting multiple data sets into a single change should be carefully planned. The export function cannot check whether the changes in the specified sequence will logically work as desired. The changes will be imported into the change individually in the sequence they are specified, and you must ensure that any change in the list logically has all preceding changes as prerequisites.

If you requested TSO mode, the requested changes are exported.

If you requested batch mode, a JCL job is generated. You must submit the job to export the changes.

Related concepts

“Change Management terminology” on page 639
Before you use Change Management (CM), you should understand the terms that CM uses.

Related reference

“Types of changes and change status” on page 644

To facilitate change management, Db2 Admin Tool categorizes changes into several types and assigns a status to each change as it moves through the change management process.

Importing changes

You can create a change by importing SQL statements from a data set. When you import the statements, a new change is created and registered. You can import multiple delta changes as a group in one change.

About this task

You can import SQL statements (DDL), delta changes that are generated by IBM Db2 Object Comparison Tool for z/OS, or a mix of SQL statements and delta changes if they logically relate.

Object Comparison Tool creates a delta change if CHANGE is specified for the **Generate apply jobs** field on the **Generate Compare Jobs (GOC5)** panel. (This specification generates parameter CMDELTA for GOC2CMP).

Warning: Importing an Object Comparison Tool change data set that is not generated as a delta change can have unwanted side effects that cannot be checked during the import process.

Requirements:

- When you import SQL statements into a change, the subsystem that is used for the import operation must support the SQL statements that you are importing.
- If you are importing a delta changes data set, the data set must represent one generated delta changes file. Concatenating or merging multiple data sets into one can cause unpredictable results, because statements are reordered during the import process.
- The data sets from which you are importing the SQL statements must be one of the following types:
  - One of the following delta changes data sets:
    - A delta changes data set that was generated by Change Management when changes were promoted from another system, as described in "Promoting changes" on page 838.
    - A delta changes data set that was generated by Object Comparison Tool when you set the **Generate apply jobs** field to Change on the **Generate Compare Jobs (GOC5)** panel. (See Generating a compare batch job (IBM Db2 Object Comparison Tool for z/OS 12.1.0).)

    Any other change data sets that are produced by Db2 Admin Tool cannot be imported.
  - A data set that contains SQL statements and meets the following requirements:
    - A fixed-block sequential data set (RECFM=F*x*)
    - A member of a partitioned data set with a logical record length of 80 (LRECL=80)

You can create a single change by importing multiple files at the same time; all data set types that are valid for import operations can be part of the same import.

During the import process, the syntax of each change statement in the imported SQL statements is checked. However, semantic checking is done during the analyze process.

Restriction: Changes to tables or sequences using alias names is not supported. The actual name of the object must be used.

Procedure

To import changes:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option 1, and press Enter.
3. On the **Manage Changes (ADB2C1)** panel, specify option 4, and press Enter.
4. On the **Import Changes (ADB2C14)** panel, specify the name of the data set that contains the SQL statements that you want to import, and press Enter.

If the input data set is a partitioned data set (PDS), you must specify a member name or a member pattern (as defined by ISPF). If a member pattern is specified, all members that fit the pattern are added to the list of data sets to import; they are listed in member name sequence.

5. Specify additional input data sets as needed:

```
ADB2C14 n ------------------ CM - Import Changes ------------------- 08:05
Command ===

Commands : CONTINUE RESET                                DB2 System: DD1A

Input data set information:
  Data set name .
    Member  . . .          (member name or pattern if partitioned)
Line commands :
M - Move   A - After  B – Browse  D - Delete

Select Seq Data set name                                                Oper.
------ --- --------------------------------------------------------     ------

         1 USER01.PROD.CHANGES(FEB08001)
         2 USER01.PROD.CHANGES(FEB08002)
         3 USER01.PROD.CHANGES(FEB08003)
         4 USER01.PROD.CHANGES(FEB08010)
         5 USER01.PROD.CHANGES(FEB08011)
         6 USER01.PROD.CHANGES(FEB08071)
         7 USER01.PROD.CHANGES(FEB08072)
         8 USER01.PROD.CHANGES(FEB08073)
         9 USER01.PROD.CHANGEXX
        10 USER01.PROD.CHANGES(XXCHGA)
        11 USER01.PROD.CHANGES(XXCHGB)
******************************* END OF DB2 DATA *******************************
```

If you need to clear the list of data sets, issue the RESET command.

6. Specify the sequence in which you want these input data sets processed by using the line commands M and A to move entries in the list.

   The input data sets are processed in the sequence in which they are listed on the panel.

   Recommendation: Carefully plan for any situation where you are importing multiple data sets into a single change. Db2 Admin Tool cannot check whether the changes in the specified sequence will logically work as desired. The input changes are imported into the change individually in the sequence that they are specified. You must ensure that any change in the list logically has all preceding changes as prerequisites.

7. Issue the NEXT command, and press Enter.
8. On the **Select process modes (ADB2C14M)** panel, specify values for **Prereq resolution mode** and **Execution mode**, and press Enter:

```
ADB2C14M  ------- CM Import changes - Select process modes --14:28


 Specify how to continue Import :

 Prereq resolution mode  . BATCH     (TSO/Batch)
 Execution mode  . . . . . BATCH     (TSO/Batch)
```

   Importing a change is a two-phase process in which Db2 Admin Tool determines if any pending changes exist for the objects and then registers the imported change. The processing modes are:

   TSO
   Perform the processing in the foreground.

   Batch
   Perform the processing in background.

9. If you specify TSO for both **Prereq resolution mode** and **Execution mode**, complete the following steps:

   a) On the **Register Options (ADB2CRO)** panel, specify the following information, and issue the NEXT command:

- An owner for the change. (The default owner is the current SQL ID.)
- A name for the change.

Optionally, you can also specify a comment, an ignore, and a mask for the change.

b) If the changes in the data set affect objects that have pending changes, on the **Import - Pending Changes (ADB2C14P)** panel, specify one of the following actions to take, and press Enter.

Prereq
Make the pending changes a prerequisite for this change.

Supersede
Make this change a prerequisite change for the pending changes. The pending changes are put in DEFINED status.

Ignore
Ignore the pending changes. The pending changes are left in analyzed status.

Virtual changes are not applied to the object and prerequisites are not established. You are responsible for establishing the prerequisites and ensuring that pending changes do not conflict with the current change. To help identify any conflicting changes, use run-time analyze when running this change.

Cancel
Cancel importing this change.

Display
Display the changes that are pending.

c) On the **Import Changes (ADB2C14)** panel, verify the message that indicates whether the change was registered successfully.

The change is put in DEFINED status. If pending changes exist and you specified `Ignore` as the action to take, pending changes are not put in DEFINED status.

10. If you specify TSO for **Prereq resolution mode** and `Batch` for **Execution mode**, complete the following steps:

a) On the **Promote Target Output Data Set Name (ADB2DSAL)** panel, specify the name of an output data set name to contain the (delta) change statements, and press Enter.

b) On the **Register Options (ADB2CRO)** panel, specify the following information, and issue the NEXT command:

- An owner for the change. (The default owner is the current SQL ID.)
- A name for the change.

Optionally, you can also specify a comment, an ignore, and a mask for the change.

c) If the changes in the data set affect objects that have pending changes, on the **Import - Pending Changes (ADB2C14P)** panel, specify one of the following actions to take, and press Enter.

Prereq
Make the pending changes a prerequisite for this change.

Supersede
Make this change a prerequisite change for the pending changes. The pending changes are put in DEFINED status.

Ignore
Ignore the pending changes. The pending changes are left in analyzed status.

Virtual changes are not applied to the object and prerequisites are not established. You are responsible for establishing the prerequisites and ensuring that pending changes do not conflict with the current change. To help identify any conflicting changes, use run-time analyze when running this change.

Cancel
Cancel importing this change.

Display
Display the changes that are pending.

d) Review the generated job to register the change and submit the JCL.

When the job completes successfully, the change is registered and put in DEFINED status. If pending changes exist and you specified `Ignore` as the action to take, pending changes are not put in DEFINED status.

11. If you specify `Batch` for **Prereq resolution mode**, you must specify `Batch` for **Execution mode**. In this case, complete the following steps:

a) On the **Import a Change - Action for Pending Changes (ADB2CONF)** panel, specify one of the following actions to take if pending changes exist for the objects that the imported change affects:

Prereq
Make the pending changes a prerequisite for this change.

Supersede
Make this change a prerequisite change for the pending changes. The pending changes are put in DEFINED status.

Ignore
Ignore the pending changes. The pending changes are left in analyzed status.

Virtual changes are not applied to the object and prerequisites are not established. You are responsible for establishing the prerequisites and ensuring that pending changes do not conflict with the current change. To help identify any conflicting changes, use run-time analyze when running this change.

Cancel
Cancel importing this change.

Recommendation: Specify `Cancel` to avoid registering the changes if pending changes exist. You can review the batch output, which lists the pending changes and decide whether to keep them as prerequisite changes or supersede them. Then, you can import the change again and specify either `Prereq` or `Supersede`.

b) On the **Register Options (ADB2CRO)** panel, specify the following information, and issue the NEXT command:

- An owner for the change. (The default owner is the current SQL ID.)
- A name for the change.

Optionally, you can also specify a comment, an ignore, and a mask for the change.

c) Review the generated job and submit the JCL.

Results

You can now display your imported change ("Displaying changes" on page 870), and then run it ("Running a change" on page 656).

Viewing the change that supersedes a change

If the status of a change switches from ANALYZED to DEFINED, displaying the superseding change might help you determine the cause of the switch in status.

About this task

When a new change is created or imported to supersede an existing change, that existing change is set to DEFINED status and needs to be analyzed again before it can be run. In this case, you might want to see which change caused the status switch by viewing the superseding change.

Procedure

To view the change that supersedes a change:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option 1, and press Enter.
3. Optional: At the bottom of the **Manage Changes (ADB2C1)** panel, specify any search criteria to filter or limit the changes that are displayed.
4. On the **Manage Changes (ADB2C1)** panel, specify option 1, and press Enter.
5. On the **CM - Changes (ADB2C11)** panel, in the list of changes, enter the SBY line command next to the change for which you want to view the superseding change:

```
ADB2C11 n ------------------ CM - Changes -------------------- Row 1 to 4 of 4
Command ===>                                                  Scroll ===> CSR

Commands: COMMENT  EXPORT  REFRESH
Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks  S - Show WSL  B - Checkpoint
 ? - Show all line commands

Sel       ID Owner    Name                        Type     Status   I Comment
           * *        *                           *        *        * *
--- -------- -------- --------------------------- -------- -------- - ---------
           1 TS5777   CHG001                      CHANGE   DEFINED    TEST1
SBY        2 TS5777   CHG002                      CHANGE   DEFINED    TEST2
           3 TS5777   CHG003                      CHANGE   DEFINED    TEST3
           4 TS5777   CHG004                      CHANGE   DEFINED    TEST4
******************************* END OF DB2 DATA *******************************
```

Tip: The SBY line command displays the change that immediately supersedes the selected change. If you want to view more prerequisite changes, use the PR line command.

The change that supersedes the selected change is displayed:

```
ADB2C11 n ------------------ CM - Changes -------------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Commands: COMMENT  EXPORT  REFRESH
Line commands:
 U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
 PQ - Prerequisites  IG - Ignores  MA - Masks  S - Show WSL  B - Checkpoint
 ? - Show all line commands

Sel       ID Owner    Name                        Type     Status   I Comment
           * *        *                           *        *        * *
--- -------- -------- --------------------------- -------- -------- - ---------
           4 TS5777   CHG004                      CHANGE   DEFINED    TEST4
******************************* END OF DB2 DATA *******************************
```

In this example CHG004 supersedes CHG002.

Related reference

"Types of changes and change status" on page 644
To facilitate change management, Db2 Admin Tool categorizes changes into several types and assigns a status to each change as it moves through the change management process.

Ignores

An *ignore* provides the ability to specify that certain fields in the Db2 catalog records are to be ignored when objects are compared. Ignores help avoid meaningless comparisons and protect those fields, called *ignore fields*, from being changed.

For example, you might want to ignore fields that contain space information, because production tables and indexes are often larger than the corresponding test tables and indexes. You might also want to ignore fields that contain buffer pool names, because a broader set of buffer pools might be implemented in the production system.

You can specify ignore fields during comparisons. Comparisons occur in the following situations:

- When you use IBM Db2 Object Comparison Tool for z/OS to compare objects
- When you run a change in Db2 Admin Tool

The place where you define ignores depends on the comparison process for which you want to use them. Ignores that are specified when running a change must be defined in the Change Management (CM) database. Ignores that are specified when using Object Comparison Tool to compare objects can be defined in either the CM database or in a data set.

When you specify an ignore, the specified fields in the Db2 catalog records are not used for comparisons. If you must re-create an object for other changes, values for ignored fields are taken from the target version. All other field values are taken from the source version.

Recommendations:

- Consider managing all your ignores through Change Management. When ignores are stored in the CM database, they are easy to track and recover because they are stored in Db2 tables.
- Use caution when specifying ignore fields. If possible, use the generic specifications, which account for some common fields that are often intentionally different on source and target systems. See "Generic ignores" on page 851.
- When specifying ignore fields, consider that many fields in theDb2catalog records are interdependent. Therefore, when one field is ignored, the value in another field might be invalid if that field is not also ignored. For example, consider the TYPE fields for tables and table spaces. If TYPE is ignored for table spaces, a table space could keep the LARGE attribute. If the compare source is a segmented table space, the resulting set of attributes is invalid if the SEGSIZE field is not also ignored. Another example of dependency is between the SQTY and SECQTYI fields in SYSTABLEPART and SYSINDEXPART. If the secondary quantity is to be ignored, specify both fields or use the generic SPACE specification.

Related information:

"Change Management (CM)" on page 638

- "System ignores" on page 847
- "Ignore fields" on page 847
- "Ignore syntax" on page 850
- "Generic ignores" on page 851
- "The Manage Ignores (ADB2C3) panel" on page 853

System ignores

Some catalog fields are automatically ignored, such as statistics, dates, and internal identifiers, because these fields are generally meaningless for comparisons. These types of ignores are called *system ignores*. These ignores are included by default and do not need to be explicitly specified on the ISPF panels. You can list system ignores by setting the compare reporting option **System generated** to YES; See Batch compare report format (IBM Db2 Object Comparison Tool for z/OS 12.1.0).

If the compared objects originate from two different versions of Db2, they might be different, because more parameters, attributes, or clauses are supported by one of the versions (typically the newer version). In this case, Object Comparison Tool ignores such fields just like ignore fields that are manually entered. No differences of this kind are reported.

Ignore fields

Only certain fields in certain Db2 catalog tables can be ignored. The following table shows the Db2 catalog tables and the fields that you can specify for ignores.

Table 38. The Db2 catalog table ignore fields

Db2 catalog table	**Ignore fields**		
SYSCHECKS	CHECKCONDITION CREATOR		
SYSCOLUMNS	ALTEREDTS COLTYPE[1] DEFAULT[1, 2] DEFAULTVALUE[1] FLDPROC[4] FOREIGNKEY[3]	KEYSEQ LABEL LENGTH[1] LENGTH2 NULLS[1] PARTKEY_COLSEQ	PARTKEY_ORDERING REMARKS SCALE[1] STATS_FORMAT TYPENAME TYPESCHEMA
SYSCONTROLS	OWNER OWNER_TYPE ENABLED	REMARKS RULETEXT	
SYSDATABASE	BPOOL CREATOR DBCS_CCSID ENCODING_SCHEME	GROUP_MEMBER INDEXBP MIXED_CCSID ROSHARE	SBCS_CCSID STGROUP TYPE
SYSDATATYPES	ENCODING_SCHEME LENGTH METATYPE	OWNER SCALE SOURCESCHEMA	SOURCETYPE SUBTYPEREMARKS
SYSFIELDS	EXITPARM EXITPARML FLDPROC	PARMLIST WORKAREA	
SYSINDEXES	AVGKEYLEN BPOOL CLOSERULE CLUSTERING COPY CURRENT_VERSION	DSETPASS ERASERULE INDEXTYPE OLDEST_VERSION PADDED PGSIZE	PIECESIZE RELCREATED REMARKS SPACEF UNIQUERULE VERSION
SYSINDEXPART	DSNUM FREEPAGE GBPCACHE INDEXTYPE LEAFFAR LEAFNEAR	LIMITKEY PARTITION PQTY PSEUDO_DEL_ENTRIES SECQTYI	SPACEFEXTENTS SQTY STORTYPE STORNAME VCATNAMEPCTFREE
SYSSYSKEYS	COLSEQ ORDERING		

Table 38. The Db2 catalog table ignore fields (continued)

Db2 catalog table	Ignore fields		
SYSPARMS	CAST_FUNCTION CCSID ENCODING_SCHEME LENGTH LOCATOR ORDINAL	OWNER PARMNAME ROWTYPE SCALE SPECIFICNAME	SUBTYPE TABLE TABLE_COLNO TYPENAME TYPESCHEMA
SYSRELS	CHECKEXISTINGDATA DELETERULE	ENFORCED IXNAME	IXOWNER RELNAME
SYSROUTINES	ASUTIME CAST_FUNCTION CLASS COMMIT_ON_RETURN DBINFO DEBUG_MODE DETERMINISTIC EXTERNAL_ACTION EXTERNAL_NAME EXTERNAL_SECURITY FENCED FINAL_CALL FUNCTION_TYPE INLINE JAR_ID JARSCHEMA	JAVA_SIGNATURE LANGUAGE LOBCOLUMNS MAX_FAILURE NULL_CALL NUM_DEP_MQTS OWNER OWNERTYPE PACKAGEPATH PARALLEL PARAMETER_CCSID PARAMETER_STYLE PARM_COUNT PARSETREE PROGRAM_TYPE REMARKS	RESULT_COLS RESULT_SETS RUNOPTS SCRATCHPAD SCRATCHPAD_LENGTH SECURE SOURCESCHEMA SOURCESPECIFIC SPECIAL_REGS SPECIFICNAME SQL_DATA_ACCESS STAYRESIDENT SYSTEM_DEFINED TEXT WLM_ENV_FOR_NESTED WLM_ENVIRONMENT
SYSSEQUENCES	CACHE CYCLE INCREMENT MAXVALUE	MINVALUE ORDER OWNER PRECISION	REMARKS RESTARTWITH SEQTYPE START
SYSTABLEPART	COMPRESS DSNUM EXTENTS FREEPAGE IXCREATOR IXNAME	LIMITKEY LOGICAL_PART PCTFREEGBPCACHE PQTY SECQTYI SPACEF	SQTY STORNAME STORTYPE TRACKMOD VCATNAME
SYSTABLES	AUDITING CHECKS CLUSTERTYPE CREATEDBY DATACAPTURE DBNAME	EDPROC ENCODING_SCHEME KEYCOLUMNS LABEL LOCATION REMARKS	STATUS TBCREATOR TBNAME TSNAME TYPE VALPROC

Table 38. The Db2 catalog table ignore fields (continued)

Db2 catalog table	Ignore fields		
SYSTABLESPACES	BPOOL CLOSERULE CREATOR DBCS_CCSID DSETPASS ENCODING_SCHEME ERASERULE	IMPLICIT INSERTALG LOCKMAX LOCKRULE MAXROWS PARTITIONS	PGSIZE SBCS_CCSID MIXED_CCSID SEGSIZE STATUS TYPE
SYSTRIGGER	GRANULARITY OWNER	REMARKS TRIGEVENT	TRIGTIME TEXTTRIGNAME
SYSVIEWS	APP_ENCODING_CCSID CHECKTEXT ENABLE ISOLATION	MAINTENANCE PATHSCHEMAS REFRESH REFRESH_TIME	RELCREATED SIGNATURE TYPE

Notes:

1. The SYSCOLUMNS fields COLTYPE, LENGTH, SCALE, DEFAULT, and DEFAULTVALUE are all part of the column type definition. The NULLS field is also related, because in some cases, it is part of the default specification. If you choose to ignore some, but not all, of the fields that are part of a column definition, the result can be inconsistent attributes and, subsequently, invalid DDL.
2. The DEFAULT field can have a relationship to a SYSSEQUENCES row. Ignoring the DEFAULT field can cause the SYSSEQUENCES row to be included or excluded, depending on the value of the DEFAULT field in the target SYSCOLUMNS row. To ignore fields in the SYSSEQUENCES row, you must explicitly select them.
3. The FOREIGNKEY field specifies the subtype of a character type column. Ignoring the FOREIGNKEY field not only removes the check for SBCS and MIXED data, but also the FOR BIT DATA specification. As a result, CCSID conversions can occur, if applicable.
4. The FLDPROC field can have a relationship to a SYSFIELDS catalog row. Ignoring the FLDPROC field can cause the SYSFIELDS row to be included or excluded, depending on the value of FLDPROC in the target SYSCOLUMNS row. To ignore fields in the SYSFIELDS row, you must explicitly select them.

Important: Some values are stored in the Db2 catalog in both internal and external formats. Internal format is only understood by Db2 (not documented); external format is suitable for input and output. Object Comparison Tool always ignores the internal format. To ignore the value, there must be an ignore specification for the field that contains the external format of the value. For example, SYSINDEXPART.LIMITKEY stores the high value of the limit key of the partition in internal format. SYSTABLEPART.LIMITKEY stores the high value of the partition in external format. Therefore, if you need to ignore SYSINDEXPART.LIMITKEY, specify SYSTABLEPART.LIMITKEY.

Ignore syntax

The syntax for specifying an ignore is:

```
objecttype: field1, field2,...,fieldn
```

where:

- *objecttype* is the Db2 catalog table name
- *fieldx* is the Db2 catalog column to be ignored

For example, the following lines show ignore field specifications. The first specification is for a database. It shows that for SYSDATABASE, the field BPOOL is ignored when the comparison is performed.

```
SYSDATABASE: BPOOL

SYSDATABASE: INDEXBP,STGROUP

SYSTABLESPACE: BPOOL

SYSTABLEPART: PQTY,SQTY,STORNAME,VCATNAME

SYSINDEXES: INDEXSPACE

SYSINDEXPART: PQTY,SQTY,STORNAME,VCATNAME
```

Generic ignores

Generic ignores specify that you want to ignore all information of a certain type, such as all buffer pools, all allocated space information, and all information about how data is stored and partitioned. Specifying a generic ignore has the same effect as specifying ignore fields individually.

The generic ignore specifications are:

- BUFFERPOOL
- BUSINESS_TIME
- COLUMN_MASKS
- HASH_ORGANIZATION
- INCLUDE_COLUMNS
- KEYTARGETS
- PARTITIONING
- PBG_NUMPARTS
- ROW_PERMISSIONS
- SOURCE_PENDING_CHANGES
- SPACE
- STORAGE
- SYSTEM_TIME
- XMLMODIFIER

For those generic ignore specifications that directly correspond to catalog table columns, the following table shows which catalog fields are ignored when the generic ignore is specified.

Table 39. Generic ignore specifications

Generic ignore	Db2 catalog table	Ignore fields
BUFFERPOOL	SYSDATABASE	BPOOL, INDEXBP
	SYSINDEXES	BPOOL
	SYSTABLESPACE	BPOOL
BUSINESS_TIME	SYSCOLUMNS	COLTYPE
		LENGTH
		SCALE
		NULLS
		DEFAULT

Table 39. Generic ignore specifications (continued)

Generic ignore	Db2 catalog table	Ignore fields
HASH_ORGANIZATION	SYSTABLES	HASHKEYCOLUMNS
	SYSCOLUMNS	HASHKEY_COLSEQ
	SYSTABLEPART	HASHSPACE
	SYSTABLESPACE	HASHSPACE
	SYSINDEXES	HASH
KEYTARGETS	SYSINDEXES	KEYTARGET_COUNT
		IX_EXTENSION_TYPE
	SYSKEYTARGETS	KEYSEQ
		ORDERING
		TYPESCHEMA
		TYPENAME
		DATATYPEID
		SOURCETYPEID
		LENGTH
		SCALE
		NULLS
		CCSID
		SUBTYPE
		DERIVED_FROM
PARTITIONING	SYSINDEXPART	PARTITION
	SYSTABLESPACE	PARTITIONS
	SYSINDEXPART	LIMITKEY
	SYSTABLEPART	LIMITKEY
		LIMITKEY_INTERNAL
		LOGICAL_PART
		PARTITION
	SYSTABLES	PARTKEYCOLNUM
	SYSCOLUMNS	PARTKEY_COLSEQ
		PARTKEY_ORDERING
	SYSAUXRELS	PARTITION
PBG_NUMPARTS	SYSTABLESPACE	PARTITIONS
SPACE	SYSINDEXPART	PQTY, SQTY, FREEPAGE, PCTFREE, SECQTYI
	SYSTABLEPART	PQTY, SQTY, FREEPAGE, PCTFREE, SECQTYI
	SYSTABLESPACE	MAXROWS

Table 39. Generic ignore specifications (continued)

Generic ignore	Db2 catalog table	Ignore fields
STORAGE	SYSDATABASE	STGROUP
	SYSINDEXPART	STORTYPE, STORNAME, VCATNAME
	SYSTABLEPART	STORTYPE, STORNAME, VCATNAME
	SYSSTOGROUP	VCATNAME
	SYSVOLUMES	VOLID
SYSTEM_TIME	SYSCOLUMNS	COLTYPE
		LENGTH
		SCALE
		NULLS
		DEFAULT
XMLMODIFIER	XSROBJECTS	XSROBJECTSCHEMA
		XSROBJECTNAME
		TARGETNAMESPACE
		SCHEMALOCATION
	SYSXMLTYPMSCHEMA	ELEMENT_NAME

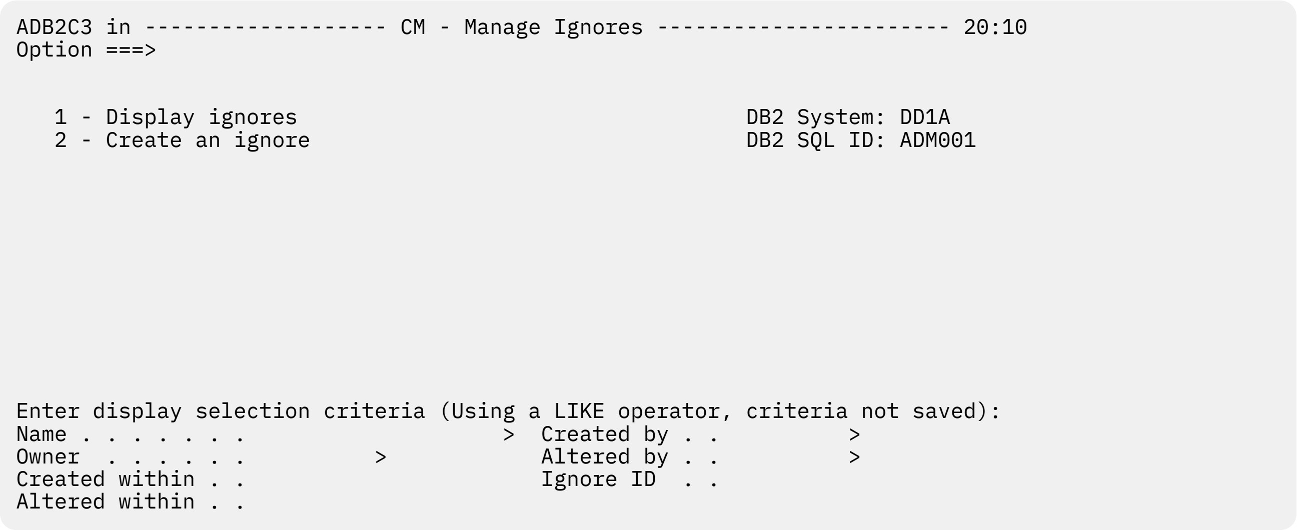
For those generic ignores that do not correspond to catalog columns, the following table describes the information that is ignored:

Table 40. Generic ignore descriptions for ignores that do not correspond to catalog tables

Generic ignore	Information that is ignored
COLUMN_MASKS	Columns masks See Column mask (Db2 12 for z/OS).
ROW_PERMISSIONS	Row permissions See Row permission (Db2 12 for z/OS).
INCLUDE_COLUMNS	Columns that are specified in the INCLUDE clause of the CREATE INDEX statement for unique indexes See CREATE INDEX (Db2 12 for z/OS).
SOURCE_PENDING_CHANGES	Pending changes on the source objects

The Manage Ignores (ADB2C3) panel

The **Manage Ignores (ADB2C3)** panel is the main menu for managing ignores in the CM database. From this panel, you can view the existing ignores or create a new ignore.



```
ADB2C3 in ------------------- CM - Manage Ignores ----------------------- 20:10
Option ===>


   1 - Display ignores                                    DB2 System: DD1A
   2 - Create an ignore                                   DB2 SQL ID: ADM001











Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .                   >  Created by . .          >
Owner  . . . . . .          >           Altered by . .          >
Created within . .                      Ignore ID  . .
Altered within . .
```

Figure 400. ***Manage Ignores (ADB2C3)*** *panel*

Displaying ignores in the Change Management database

Ignores can be stored either in the Change Management (CM) database or, if you are using them with IBM Db2 Object Comparison Tool for z/OS, in a data set.

About this task

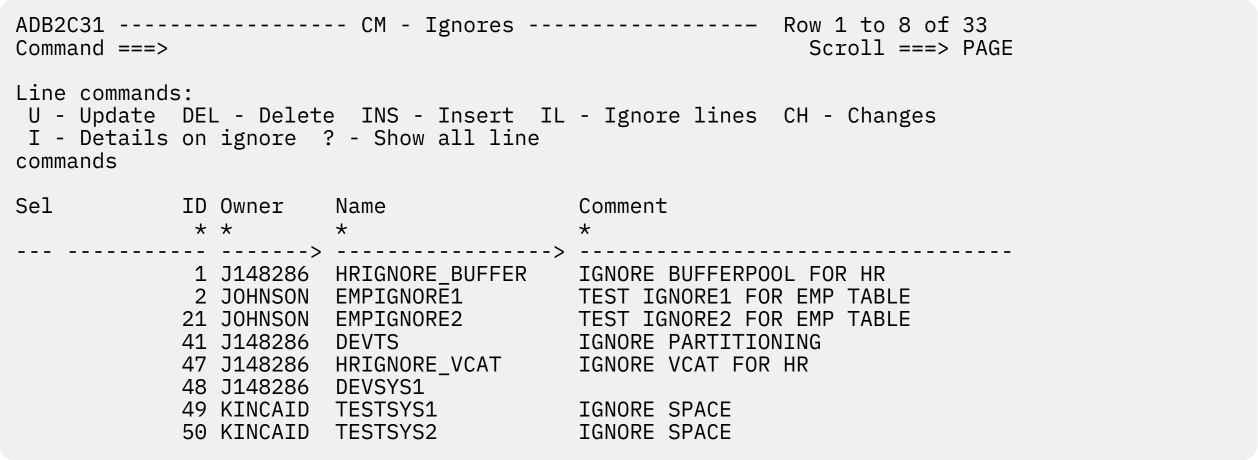
This procedure does not apply to ignores that have been created in an explicitly named data set outside of Change Management. For information about how to create, view, and edit ignores in a data set, see Specifying ignores (IBM Db2 Object Comparison Tool for z/OS 12.1.0).

Procedure

To display ignores in the CM database:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option **CM**, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option **3**, and press Enter.
3. Optional: At the bottom of the **Manage Ignores (ADB2C3)** panel, specify search criteria to filter the ignores that you want to display.
4. On the **Manage Ignores (ADB2C3)** panel, specify option **1**, and press Enter.

   The ignores in the CM database are displayed:



```
ADB2C31 ------------------ CM - Ignores ------------------  Row 1 to 8 of 33
Command ===>                                                  Scroll ===> PAGE

Line commands:
 U - Update  DEL - Delete  INS - Insert  IL - Ignore lines  CH - Changes
 I - Details on ignore  ? - Show all line
commands

Sel          ID Owner    Name               Comment
              * *        *                  *
--- ---------- -------> -----------------> ----------------------------------
              1 J148286  HRIGNORE_BUFFER    IGNORE BUFFERPOOL FOR HR
              2 JOHNSON  EMPIGNORE1         TEST IGNORE1 FOR EMP TABLE
             21 JOHNSON  EMPIGNORE2         TEST IGNORE2 FOR EMP TABLE
             41 J148286  DEVTS              IGNORE PARTITIONING
             47 J148286  HRIGNORE_VCAT      IGNORE VCAT FOR HR
             48 J148286  DEVSYS1
             49 KINCAID  TESTSYS1           IGNORE SPACE
             50 KINCAID  TESTSYS2           IGNORE SPACE
```

Figure 401. ***Ignores (ADB2C31)*** *panel*

On this panel, you can issue line commands to perform the following actions for any listed ignore:

- See the definition of the ignore and modify it (by using the IL line command)
- View details about who created the ignore and when and who altered it last (by using the I line command)
- See which changes use the ignore (by using the CH line command)
- Insert, delete, or update an ignore (by using the INS, DEL, and U line commands)

Creating ignores in the Change Management database

You can define ignores in either the Change Management (CM) database or, if you are using them with IBM Db2 Object Comparison Tool for z/OS, in a data set.

About this task

This procedure does not explain how to create ignores in an explicitly named data set outside of Change Management. For information about how to create ignores in a data set, see Specifying ignores (IBM Db2 Object Comparison Tool for z/OS 12.1.0).

Procedure

To create an ignore in the CM database:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option **CM**, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option **3**, and press Enter.
3. On the **Manage Ignores (ADB2C3)** panel, specify option **2**, and press Enter.
4. On the **Insert Ignore (ADB2C22)** panel, specify an owner and name for the ignore and optionally a comment, and press Enter.

   A message confirms that an INSERT statement was executed. That statement inserted a row with the specified name into a table in the CM repository.
5. Press F3 to return to the **Manage Ignores (ADB2C3)** panel.
6. Specify option **1**, and press Enter.
7. On the **Ignores (ADB2C31)** panel, issue the IL line command for the ignore that you just created, and press Enter.

   The **Specify Ignore Fields: Objects (GOCCI)** panel is displayed:

```
 ---------------- Specify Ignore Fields: Objects ----------- Row 1 to 18 of 18
Command ===>                                                  Scroll ===> PAGE

Valid line commands are:
 U - Update Ignore Fields

Select Object          Ignore Fields
       *               *
------ --------------- --------------------------------------------------------
       GENERIC         None
       SYSCHECKS       None
       SYSCOLUMNS      None
       SYSCONTROLS     None
       SYSDATABASE     None
       SYSDATATYPES    None
       SYSFIELDS       None
       SYSINDEXES      None
       SYSINDEXPART    None
       SYSKEYS         None
       SYSPARMS        None
       SYSPACKAGE      None
       SYSRELS         None
       SYSROUTINES     None
       SYSSEQUENCES    None
       SYSSTOGROUP     None
       SYSTABLEPART    None
       SYSTABLES       None
       SYSTABLESPACE   None
       SYSTRIGGERS     None
       SYSVIEWS        None
       SYSVOLUMES      None
       XMLMODIFIER     None
```

*Figure 402. **Specify Ignore Fields: Objects (GOCCI)** panel*

This panel lists the Db2 catalog tables for which you can define ignore fields. It also lists a GENERIC object from which you can select generic ignores. (See "Generic ignores" on page 851.)

8. Issue the U line command next to the catalog table (or GENERIC line) from which you want to select fields to ignore, and press Enter.
9. On the **Select Ignore Fields (GOCCIF)** panel, use the S line command to select a field or fields to be ignored, and press Enter.
10. Press F3 to return to the **Specify Ignore Fields: Objects (GOCCI)** panel.

    The selected fields are listed in the **Ignore Fields** column.
11. Repeat steps "8" on page 856 through "10" on page 856 as needed to specify all the ignore fields that you want for the ignore.
12. Press F3 until the **Change Management (CM) (ADB2C)** panel is displayed.

Editing and deleting ignores in the Change Management database

You can add and delete fields from an existing ignore.

About this task

This procedure does not apply to ignores that have been created in an explicitly named data set outside of Change Management (CM). For information about how to view and edit ignores in a data set, see Specifying ignores (IBM Db2 Object Comparison Tool for z/OS 12.1.0).

Procedure

To edit and delete ignores in the CM database:

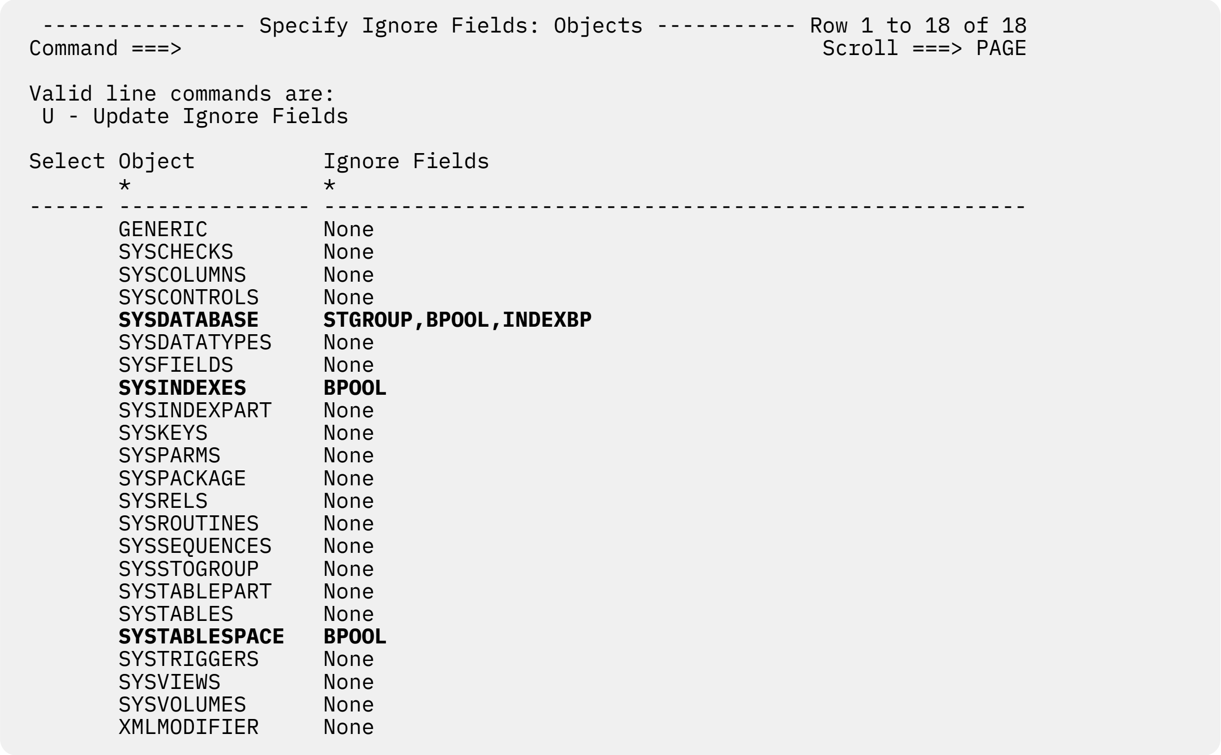
1. On the **DB2 Administration Menu (ADB2)** panel, specify option **CM**, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option **3**, and press Enter.
3. On the **Manage Ignores (ADB2C3)** panel, specify option **1**, and press Enter.

   The **Ignores (ADB2C31)** panel displays a list of ignores.

4. **If you want to delete an ignore:** Issue the DEL line command for the ignore that you want to delete, and press Enter.
5. **If you want to edit an ignore:**
   a) Issue the IL line command for the ignore that you want to edit.

   The **Specify Ignore Fields: Objects (GOCCI)** panel lists the Db2 catalog tables and the columns that are currently selected as ignore fields. (It also lists a GENERIC object and any selected generic ignores. See "Generic ignores" on page 851.)

   The following example panel shows that these fields will be ignored: the STGROUP, BPOOL, and INDEXPB fields in SYSDATABASE and the BPOOL field in SYSINDEXES and SYSTABLESPACES.



```
 ---------------- Specify Ignore Fields: Objects ----------- Row 1 to 18 of 18
Command ===>                                                Scroll ===> PAGE

Valid line commands are:
 U - Update Ignore Fields

Select Object          Ignore Fields
       *               *
------ -------------- -------------------------------------------------------
       GENERIC         None
       SYSCHECKS       None
       SYSCOLUMNS      None
       SYSCONTROLS     None
       SYSDATABASE     STGROUP,BPOOL,INDEXBP
       SYSDATATYPES    None
       SYSFIELDS       None
       SYSINDEXES      BPOOL
       SYSINDEXPART    None
       SYSKEYS         None
       SYSPARMS        None
       SYSPACKAGE      None
       SYSRELS         None
       SYSROUTINES     None
       SYSSEQUENCES    None
       SYSSTOGROUP     None
       SYSTABLEPART    None
       SYSTABLES       None
       SYSTABLESPACE   BPOOL
       SYSTRIGGERS     None
       SYSVIEWS        None
       SYSVOLUMES      None
       XMLMODIFIER     None
```

   Figure 403. Example of the definition of an ignore

   b) Issue the U line command next to the Db2 catalog table (or GENERIC line) for which you want to update the ignore fields, and press Enter.
   c) On the **Select Ignore Fields (GOCCIF)** panel, use the U and S line commands to select or de-select a particular field to be ignored, and press Enter.
   d) Press F3 to return to the **Specify Ignore Fields: Objects (GOCCI)** panel.

   The selected fields are listed in the **Ignore Fields** column.

   e) Repeat steps "5.b" on page 857 through "5.d" on page 857 as needed to specify all the ignore fields that you want for the ignore.
6. Press F3 until the **Change Management (CM) (ADB2C)** panel is displayed.

Ignore changes

An *ignore change* is a specified change to an object that you want to ignore during the compare process. Ignore changes are reported, but no SQL statements are generated for the changes.

An *ignore changes specification* is a specified list of object changes from saved compare results to be ignored in subsequent compare processing. To create and modify the contents of an ignore change specification, you must use IBM Db2 Object Comparison Tool for z/OS. See Managing ignore changes specifications (IBM Db2 Object Comparison Tool for z/OS 12.1.0). You can use Db2 Admin Tool to view, modify, or delete ignore change specifications.

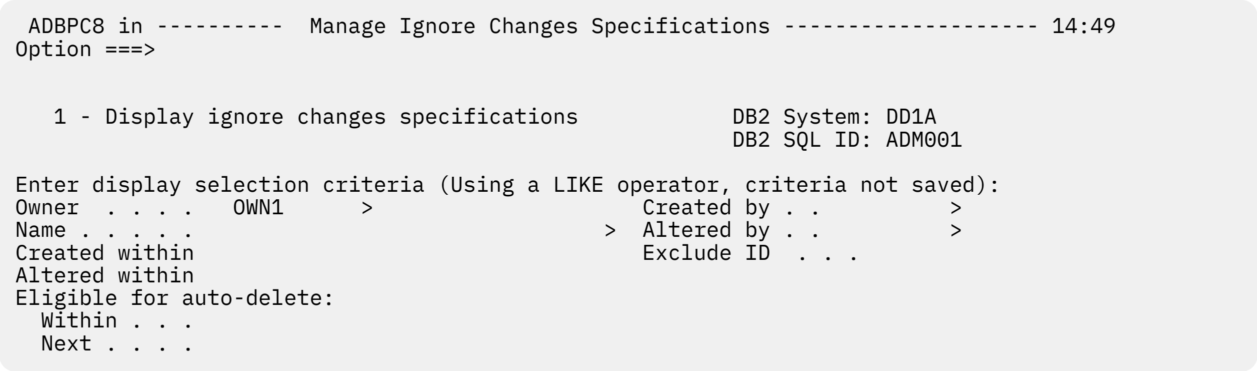
Managing ignore changes

You can use Db2 Admin Tool to view, modify, or delete ignore change specifications. To modify the contents of an ignore change specification, you must use Object Comparison Tool.

Procedure

To manage ignore changes:

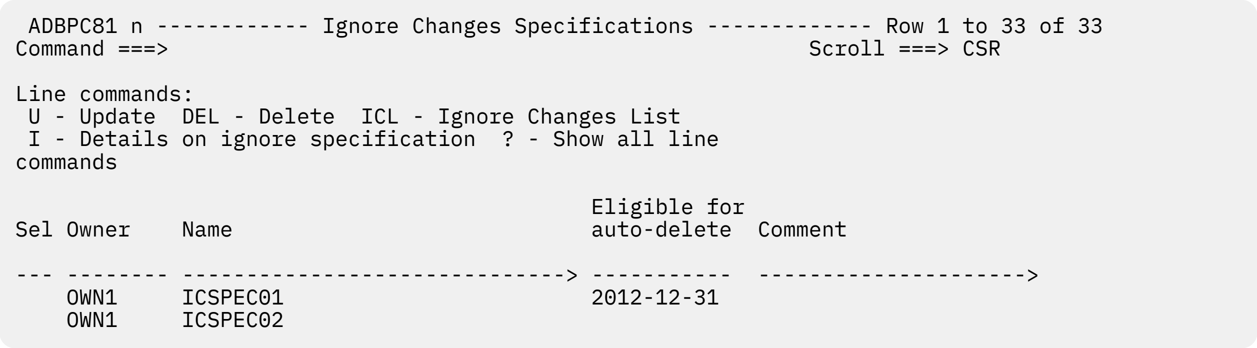
1. On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option 8 (Manage ignore changes specifications), and press Enter.
3. On the **Manage Ignore Changes Specifications (ADBPC8)** panel, specify any selection criteria, such as **Owner** and **Name** values, to identify the ignore changes specification or specifications.



```
 ADBPC8 in ----------  Manage Ignore Changes Specifications -------------------- 14:49
Option ===>


   1 - Display ignore changes specifications            DB2 System: DD1A
                                                        DB2 SQL ID: ADM001

Enter display selection criteria (Using a LIKE operator, criteria not saved):
Owner  . . . .   OWN1      >                     Created by . .          >
Name . . . . .                                >  Altered by . .          >
Created within                                   Exclude ID  . . .
Altered within
Eligible for auto-delete:
  Within . . .
  Next . . . .
```

*Figure 404. **Manage Ignore Changes Specifications (ADBPC8)** panel*

4. Specify option 1 (Display ignore changes specifications), and press Enter.
5. On the **Ignore Changes Specifications (ADBPC81)** panel, use the listed line commands to update, delete, or view more detail for one of the listed specifications.



```
 ADBPC81 n ------------ Ignore Changes Specifications ------------- Row 1 to 33 of 33
Command ===>                                                  Scroll ===> CSR

Line commands:
 U - Update  DEL - Delete  ICL - Ignore Changes List
 I - Details on ignore specification  ? - Show all line
commands

                                             Eligible for
Sel Owner    Name                            auto-delete  Comment

--- -------- ------------------------------> -----------  --------------------->
    OWN1     ICSPEC01                        2012-12-31
    OWN1     ICSPEC02
```

*Figure 405. **Ignore Changes Specifications (ADBPC81)** panel*

Related information

Managing ignore changes specifications (IBM Db2 Object Comparison Tool for z/OS 12.1.0)

Creating and managing exclude specifications

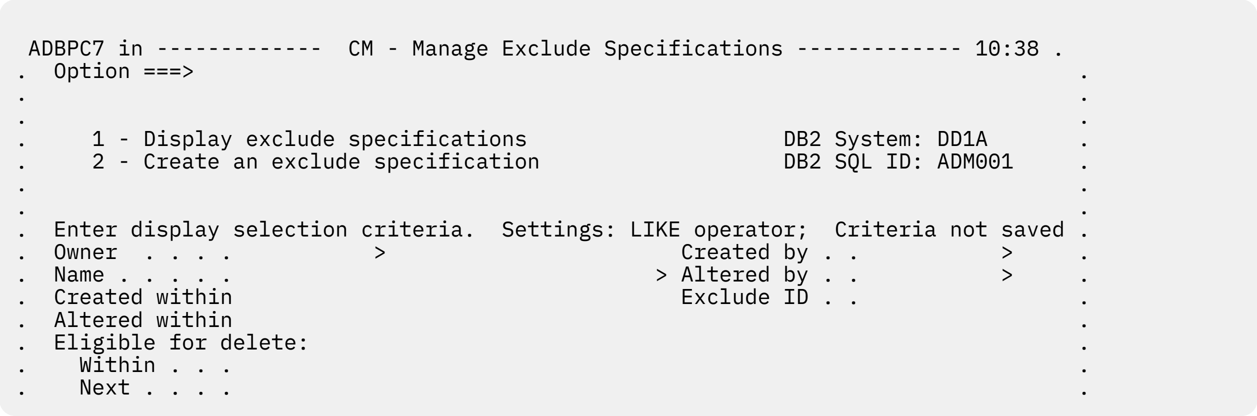
You can exclude certain objects from input and output compare processes by creating exclude specifications. An *exclude specification* is a defined list of objects to exclude from a comparison.

Procedure

To create and manage exclude specifications:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.

2. On the **Change Management (CM) (ADB2C)** panel, specify option 7 (Manage exclude specifications), and press Enter.
3. On the **Manage Exclude Specifications (ADBPC7)** panel, select option 1 to view or edit an existing specification or option 2 to create a new specification. If you specify option 1, you can optionally specify any selection criteria at the bottom of the panel:



```
 ADBPC7 in -------------  CM - Manage Exclude Specifications ------------- 10:38 .
.  Option ===>                                                                     .
.                                                                                  .
.                                                                                  .
.     1 - Display exclude specifications                    DB2 System: DD1A       .
.     2 - Create an exclude specification                   DB2 SQL ID: ADM001     .
.                                                                                  .
.                                                                                  .
.  Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved .
.  Owner  . . . .           >                       Created by . .           >     .
.  Name . . . . .                                 > Altered by . .           >     .
.  Created within                                   Exclude ID . .                 .
.  Altered within                                                                  .
.  Eligible for delete:                                                            .
.    Within . . .                                                                  .
.    Next . . . .                                                                  .
```

Figure 406. ***Manage Exclude Specifications (ADBPC7)*** *panel*

4. Press Enter
5. If you selected option 2 (Create an exclude specification), complete the following steps to create a new exclude specification:

   a) On the **CM - Insert Exclude Specification (ADB2C22)** panel, specify the owner name and specification name. You can optionally specify an **Eligible for auto-delete** value. Press Enter.

   The following message confirms that the new specification was added:

   ```
   INSERT stmt executed
   ```

   b) Exit (PF3) back to **Manage Exclude Specifications (ADBPC7)** panel.

   c) Specify option 1 (Display exclude specifications). Optionally specify any selection criteria at the bottom of the panel to help find your newly inserted specification. Press Enter.
6. On the **Exclude Specifications (ADBPC71)** panel, enter the ESL line command next to a listed specification.
7. On the **CM - Exclude Objects (ADBPC7L)** panel, use the line commands to edit the list of objects in the selected exclude specification. Type in object names and other information.
8. Exit (PF3) back to the **Manage Exclude Specifications (ADBPC7)** panel.

Related information

Excluding objects from the compare process (IBM Db2 Object Comparison Tool for z/OS 12.1.0)

Versions

A *version* is a snapshot of the definitions of a set of objects at a point in time.

The object definitions typically represent an application or application area.

Versions enable you to track the changes to a set of objects, restore objects to a previous version if you need to fall back, and promote changes from one system to another.

Versions can be created in one of three ways:

- When using Change Management, you can define a version scope (the objects to be included in a version) and then use the GV line command on the Version Scopes (ADB2C42) to generate a version based on that scope.
- When you run a change using Change Management, you can specify to have a version of the objects generated after the changes have been applied.

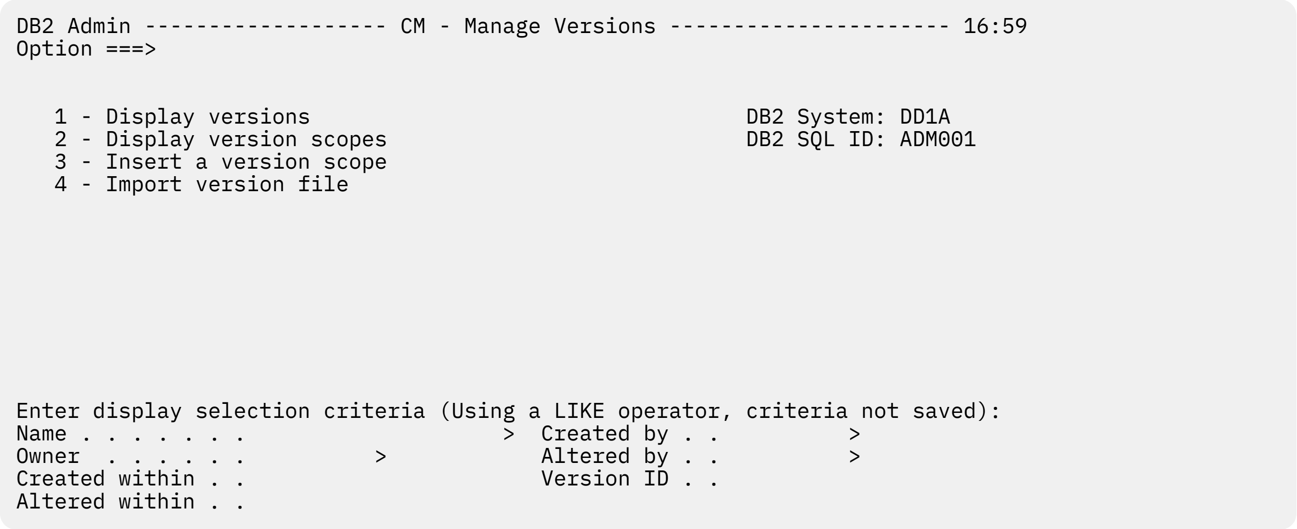
- When you use IBM Db2 Object Comparison Tool for z/OS, you can have versions of the source and target objects generated. When Change Management is enabled, you have the option of storing versions in data sets or in the Change Management database.

  Tip: Consider storing all of your versions in the Change Management database, which makes them easier to track, access, and recover.

When you promote a set of changes from one system to another, you need two versions. The *starting version* represents the state of objects before any changes are made and the *ending version* represents the state of objects after the promoted changes are made. During the promote process, Db2 Admin Tool compares the ending version with the starting version and generates a delta changes data set the contains the SQL statements that are required to bring the other system up to the same level as the system from which your promoting the changes. You can then import the delta changes data set into a new change on the system to which you are promoting the changes, analyze the change, and run them.

When you implement them carefully, you can also use versions as the base version for subsequent changes to a set of objects. When you analyze a change, Db2 Admin Tool needs a base set of definitions for the change for the analyze process. Db2 Admin Tool either extracts the object definitions from the catalog to use as the base version, which can be time consuming, or uses an existing version as the base version. You can specify that Db2 Admin Tool uses an existing version when there are no prerequisite changes for the objects.

The **CM - Manage Versions** panel, which is shown in the following figure, is the main panel for managing versions:



```
DB2 Admin ------------------- CM - Manage Versions ---------------------- 16:59
Option ===>


   1 - Display versions                                 DB2 System: DD1A
   2 - Display version scopes                           DB2 SQL ID: ADM001
   3 - Insert a version scope
   4 - Import version file









Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .                    >  Created by . .          >
Owner  . . . . . .          >            Altered by . .          >
Created within . .                       Version ID . .
Altered within . .
```

Figure 407. Manage Versions panel (ADB2C4)

Versions that have been generated in explicitly named data sets when you use IBM Db2 Object Comparison Tool for z/OS are not displayed because they are not stored in the Change Management database. When you use IBM Db2 Object Comparison Tool for z/OS and Change Management is enabled, you have the option of storing versions in data sets or in the Change Management database.

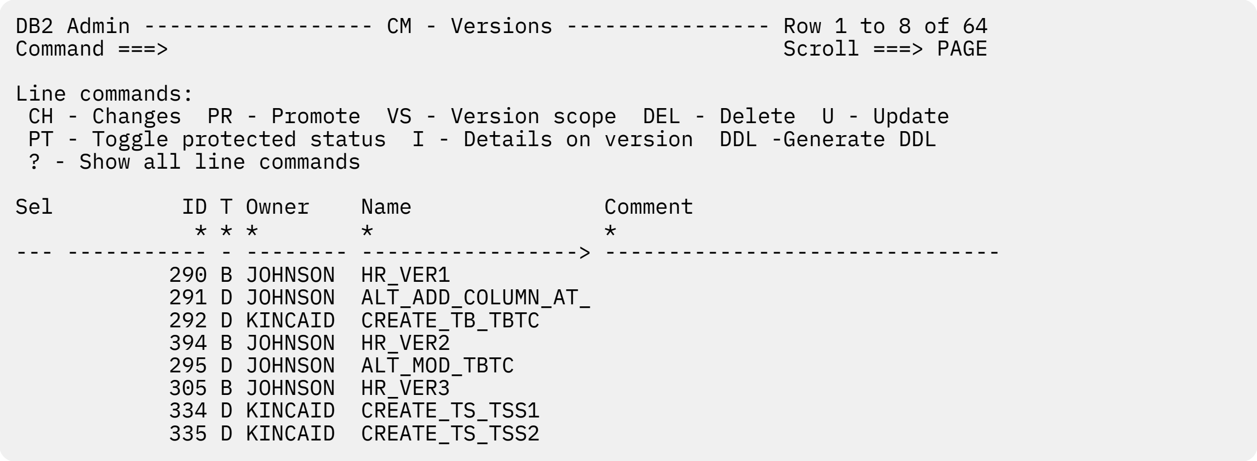
Displaying versions

You can display the versions that are stored in the Change Management database.

Procedure

1. Specify option 4 on the **Change Management (CM)** panel to display the **Manage Versions** panel.
2. Optional: Enter the search criteria to filter or limit the versions that are displayed.
3. Specify option 1 to display the Versions panel.

   The following figure shows an example of the **Versions** panel.



```
DB2 Admin ------------------ CM - Versions ---------------- Row 1 to 8 of 64
Command ===>                                                Scroll ===> PAGE

Line commands:
 CH - Changes  PR - Promote  VS - Version scope  DEL - Delete  U - Update
 PT - Toggle protected status  I - Details on version  DDL -Generate DDL
 ? - Show all line commands

Sel          ID T Owner    Name               Comment
              * * *        *                  *
--- ----------- - -------- ----------------> ------------------------------
            290 B JOHNSON  HR_VER1
            291 D JOHNSON  ALT_ADD_COLUMN_AT_
            292 D KINCAID  CREATE_TB_TBTC
            394 B JOHNSON  HR_VER2
            295 D JOHNSON  ALT_MOD_TBTC
            305 B JOHNSON  HR_VER3
            334 D KINCAID  CREATE_TS_TSS1
            335 D KINCAID  CREATE_TS_TSS2
```

Figure 408. Versions panel (ADB2C41)

Results

You can issue a variety of line commands on the Versions panel for each version. Commands are available to:

- See the changes that are associated with the version
- Promote the version
- See which scopes are associated with the version
- Set the protected status for the version
- Delete or update a version
- View details about the version

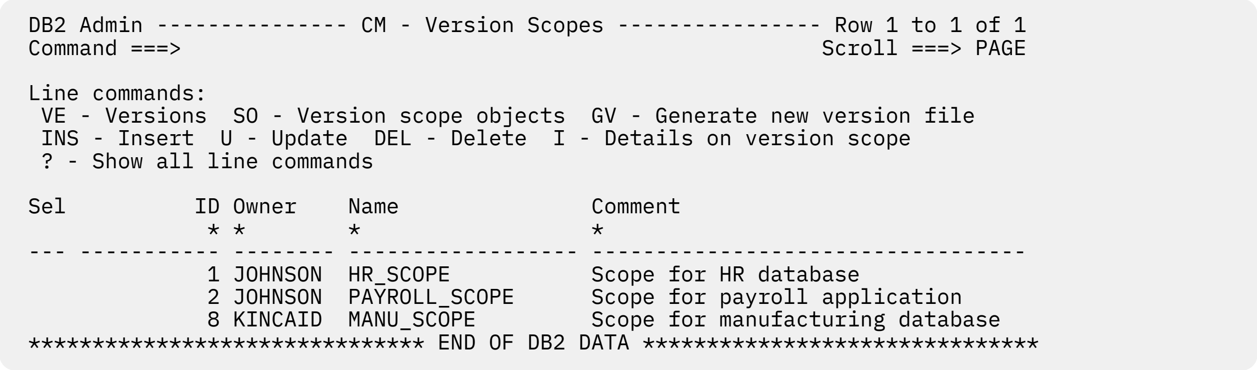
Versions that have been generated in explicitly named data sets when you use IBM Db2 Object Comparison Tool for z/OS are not displayed because they are not stored in the Change Management database. When you use IBM Db2 Object Comparison Tool for z/OS and Change Management is enabled, you have the option of storing versions in data sets or in the Change Management database.

Creating a version from a version scope

You can create a version that is stored in the Change Management database from a version scope.

Procedure

1. Specify option 4 on the **Change Management (CM)** panel to display the **Manage Versions** panel.
2. Specify option 2 to display the Version Scopes panel, as shown in the following figure:



```
DB2 Admin --------------- CM - Version Scopes ---------------- Row 1 to 1 of 1
Command ===>                                                Scroll ===> PAGE

Line commands:
 VE - Versions  SO - Version scope objects  GV - Generate new version file
 INS - Insert  U - Update  DEL - Delete  I - Details on version scope
 ? - Show all line commands

Sel          ID Owner    Name               Comment
              * *        *                  *
--- ----------- -------- ------------------ ---------------------------------
              1 JOHNSON  HR_SCOPE           Scope for HR database
              2 JOHNSON  PAYROLL_SCOPE      Scope for payroll application
              8 KINCAID  MANU_SCOPE         Scope for manufacturing database
******************************* END OF DB2 DATA *******************************
```

Figure 409. Version Scopes panel (ADB2C42)

3. Specify the GV line command for the version scope for which you want to generate a version.

4. Specify an owner and name for the new version on the pop-up panel that is displayed.

   The JCL to create the version is displayed.
5. Review and submit the job to create the new version.

Creating a version when running a change

When you run a change, you can specify that a new base version is generated. The base version can be created before or after the change is implemented.

When you create a version, you must specify the method that is used to define the content of the base version:

AUTO
: Specify AUTO if you want the product to automatically determine the objects to put into the base version based on the objects that are being changed.

USER
: Specify USER if you want to provide a version scope that defines the object list. If you specify USER, ensure that an appropriate version scope for the version to be created exists.

You can use Db2 Admin Tool online or CM batch mode to create a version when running a change.

Creating a version online

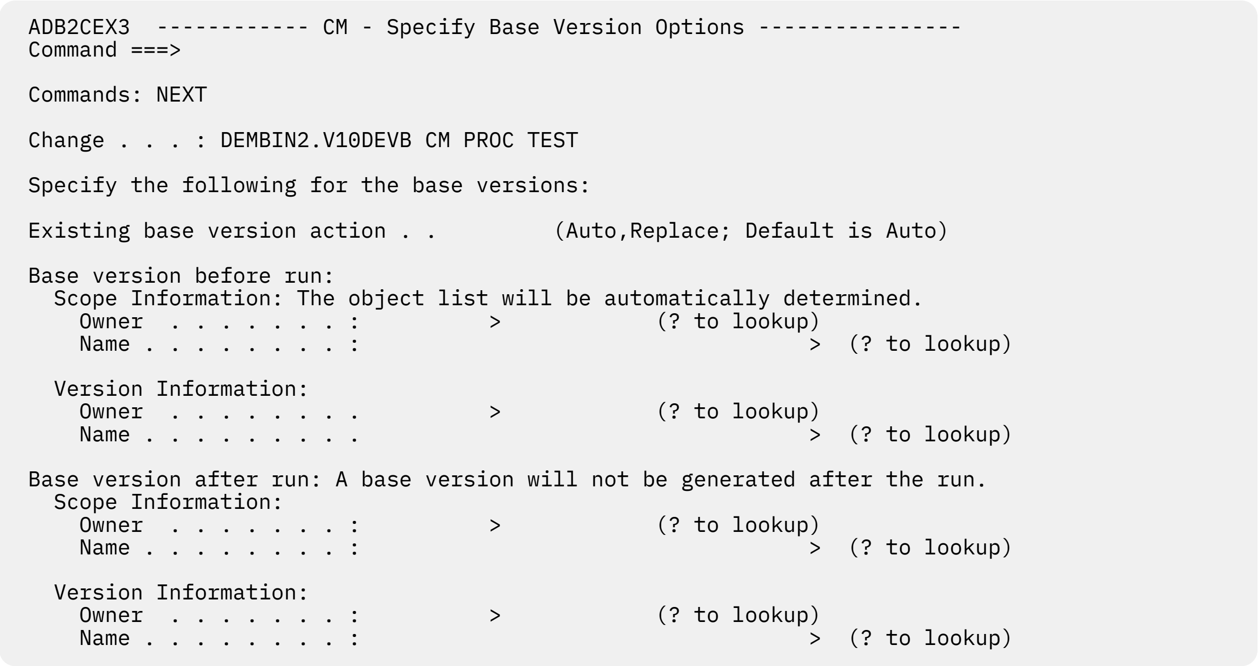
You can use the Db2 Admin Tool online interface to create a version when running a change.

Procedure

1. Display the change to be run by selecting option 1 on the **Change Management** panel, and then select option 1 on the **Manage Changes** panel.
2. Issue the RN line command for the change that you want to run. When you issue the RN line command to run the change, specify the appropriate information on the **Run a Change** panel:

   - AUTO or USER in the **Generate base version before run** field to generate a new base version immediately before the change is implemented.
   - AUTO or USER in the **Generate base version after run** field to generate a new base version immediately after the change is implemented.

   The **CM - Specify Base Version Options** panel (ADB2CEX3) is displayed after the **Run a Change** panel. In the following example, AUTO was chosen for the **Generate base version before run** option, and no base version was requested for the **Generate base version after run** option.



```
ADB2CEX3  ------------ CM - Specify Base Version Options ----------------
Command ===>

Commands: NEXT

Change . . . : DEMBIN2.V10DEVB CM PROC TEST

Specify the following for the base versions:

Existing base version action . .         (Auto,Replace; Default is Auto)

Base version before run:
  Scope Information: The object list will be automatically determined.
    Owner  . . . . . . . :          >            (? to lookup)
    Name . . . . . . . . :                                   >  (? to lookup)

  Version Information:
    Owner  . . . . . . . .          >            (? to lookup)
    Name . . . . . . . . .                                   >  (? to lookup)

Base version after run: A base version will not be generated after the run.
  Scope Information:
    Owner  . . . . . . . :          >            (? to lookup)
    Name . . . . . . . . :                                   >  (? to lookup)

  Version Information:
    Owner  . . . . . . . :          >            (? to lookup)
    Name . . . . . . . . :                                   >  (? to lookup)
```

Figure 410. CM - Specify Base Version Options

Attention: The base version will be overwritten if REPLACE is specified for the **`Existing base version action`** option. Specifying the base version owner and name is optional.

Creating a version using CM batch

You can use Db2 Admin Tool change management batch mode to create a version when running a change.

Procedure

1. Modify the JCL template, setting parameters as appropriate for the type of version that you require.

   Specify the appropriate information in the **generate_base_version_before_run** and **generate_base_version_after_run** lines:

   - AUTO or USER in the **generate_base_version_before_run** line to generate a new base version immediately before the change is implemented.
   - AUTO or USER in the **generate_base_version_after_run** line to generate a new base version immediately after the change is implemented.

   The following JCL example imports a change, analyzes the change, and runs the change. A base version is created before and after the change is run. The base versions will be associated with the change.

```
//BASEVF   JOB (DBA123,ICE,ICE,ICE),'SAMPLE',CLASS=B,
//   MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=DBA123,TIME=(,30),
//   REGION=0M
//*
/*JOBPARM S=SY4A
//*
//LSCLIBS JCLLIB ORDER=ADB.DEVCUST.SAMP
//*
//GOCCM    EXEC GOCCM,SSID=DSNA,PLAN=ADB
//GOCCM.PARMS DD *
  change_name = 'S22957'
  ACTION_RUN_CHANGE = 'Y'
  generate_base_version_before_run = 'auto'
  generate_base_version_after_run = 'auto'
/*
//IMCHG001  DD *
--
  ALTER TABLE SCH123.EMP
```

```
    ADD COLUMN NEWCOL INT NOT NULL WITH DEFAULT;
/*
```

2. Run the JCL.

Generating DDL for the objects in a base version

You can generate DDL from a base version that is stored in Change Management.

Procedure

1. Display a list of base versions by using any of the following methods:
   - Enter the VE line command on a change to display a list of versions that are associated with the change (Admin option CM, 1, 1), and then issue the VE line command.
   - Enter the VE line command on a version scope to display a list of base versions that were created from the version scope (Admin option CM, 4, 2), and then issue VE line command.
   - Use Admin tool option CM, 4, 1 to display a list of versions.
2. Specify the DDL line command on the **CM Versions** panel to generate DDL for the objects in the base version, as shown in the following example:

```
ADB2C41 n ------------------ CM - Versions ------------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR

Line commands:
 CH - Changes  PR - Promote  VS - Version scope  DEL - Delete  U - Update
 PT - Toggle protected status  I - Details on version  DDL - Generate DDL
 ? - Show all line commands

Sel          ID T Owner    Name                           Comment
              * * *        *                              *
--- ----------- - -------- ------------------------------ --------------------
           3035 D DEMBIN2  SAMPLE
DDL        3037 B DEMBIN2  PRE-RUN 01
           3038 B DEMBIN2  POST-RUN 01
******************************* END OF DB2 DATA *******************************
```

   This DDL line command is valid only for base versions (type=B) and not delta versions (type=D).

3. The **CM Base Version DDL** panel (ADB2C41E) is displayed with the base version owner and name fields filled in.

```
ADB2C41E  ---------------- CM - Base Version DDL ----------------------------
Command ===>

Specify the following options:
Base version:
  Owner  . . . . . . . . . DEMBIN2  >            (? to lookup)
  Name . . . . . . . . . . PRE-RUN 01                     >  (? to lookup)

SQL output data set:
  Prefix for data sets . . DEMBIN2
  Data set name  . . . . . BACKUP.DDL.PRERUN01
```

Regenerating Change Management versions containing LOBs

A new version of the records layout is created if LOB objects are involved in a change management job.

About this task

This layout is not compatible with previous versions containing LOBs. Therefore, you must regenerate older versions that contain LOB columns. You can identify which change management base versions are affected by using this query:

```
SELECT OWNER,NAME,TYPE
   FROM ADB.ADBCVERSION V
     WHERE TYPE='B'
         AND EXISTS(
                   SELECT VERSIONID
                       FROM ADB.ADBCVERLINES VL
                       WHERE V.VERSIONID=VL.VERSIONID
                             AND VL.PREFIXGROUP=52)
```

You can identify the active CM changes that are affected by using this query:

```
SELECT C.OWNER,C.NAME,C.STATUS
   FROM ADB.ADBCVERSION V,ADB.ADBCHG C
     WHERE C.STATUS NOT IN ('COMPLETE','CANCELLED')
          AND V.TYPE='D'
          AND C.DELTAVERID =V.VERSIONID
          AND EXISTS(
                     SELECT VERSIONID
                        FROM ADB.ADBCVERLINES VL
                        WHERE V.VERSIONID=VL.VERSIONID
                            AND  C.DELTAVERID=VL.VERSIONID
                            AND VL.PREFIXGROUP=52)
```

To regenerate change management versions:

Procedure

1. Use the RST line command to restart INITIAL changes.
2. Make sure that RUNNING changes are completed.
3. Use the ST line command for all of the other changes that are listed and edit and SAVE one statement (without making any changes).

   The change is put into defined status and the change can be handled as usual.

Deleting versions

This procedure explains how to delete versions.

About this task

You cannot delete delta versions but you can delete base versions.

Procedure

1. Specify option 4 on the **Change Management (CM)** panel to display the **Manage Versions** panel.
2. Specify option 1 to display the **Versions** panel.
3. Issue the DEL line command for the version that you want to delete.
4. If you receive a message that indicates that the version is protected, issue the PT line command to remove the protected status and issue the DEL line command again.

   Delete the version only if you know that it is no longer needed.

Version scopes

A version scope defines the set of objects to include in the processing of a version.

A version scope determines the objects that are included in a version.

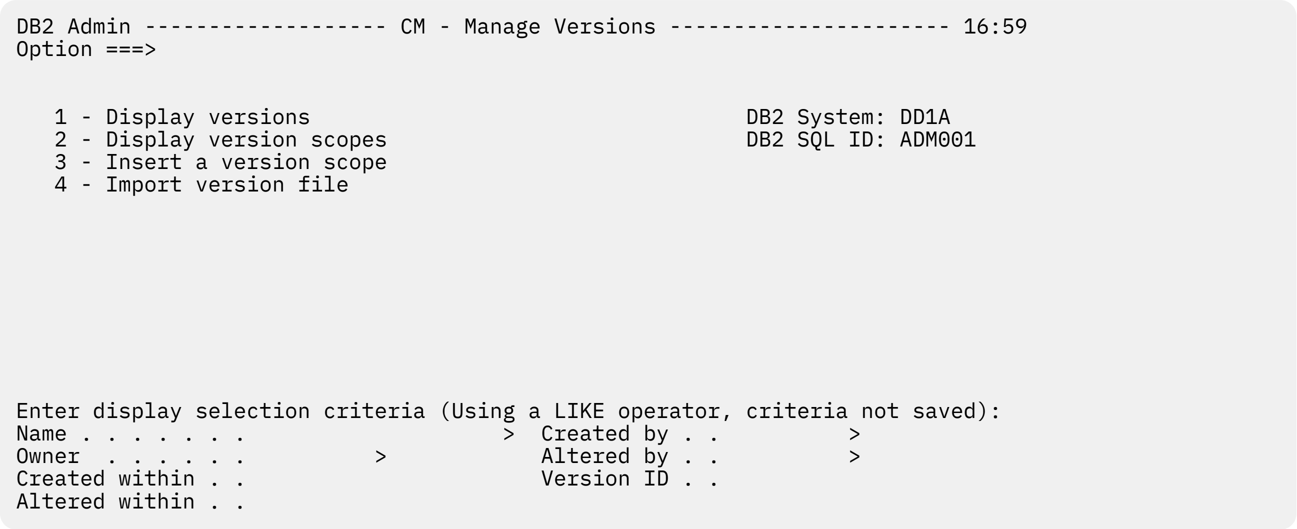
A version scope can be any set of objects, such as one or more databases, or a group of table spaces. Typically, you want to define scopes that identify all of the objects for an application or application area. For example, the scope for a human resources application should contain all the human resource databases.

After you create a version scope, you can create a base version for that set of objects.

A version scope must exist if you plan to create a new base version when you apply changes. If you have a new base version created when you run a change to reflect the object definitions after the changes, you must specify the version scope for the version.

Maintaining a version scope is a manual process, and you should ensure that the definition of the scope always includes all of the objects that you intend. For example, assume that you defined version scope SCOPE1 to include databases DB01 and DB02 and then created version BASE1. Later, you run CHANGE1, which creates a table in DB01 and creates a new database DB03, specifying to create a new base version BASE1 using SCOPE1. Database DB03 is not automatically added to SCOPE1.

The **Manage Versions** panel, which is shown in the following figure, is the main panel for working with version scopes:



```
DB2 Admin ------------------- CM - Manage Versions ---------------------- 16:59
Option ===>


   1 - Display versions                                  DB2 System: DD1A
   2 - Display version scopes                            DB2 SQL ID: ADM001
   3 - Insert a version scope
   4 - Import version file









Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .                    >  Created by . .         >
Owner  . . . . . .          >            Altered by . .         >
Created within . .                       Version ID . .
Altered within . .
```

Figure 411. Manage Versions panel (ADB2C4)

From the **Manage Versions** panel, you can display the existing version scopes to work with them or create a new version scope.

Creating version scopes

You can create a version scope.

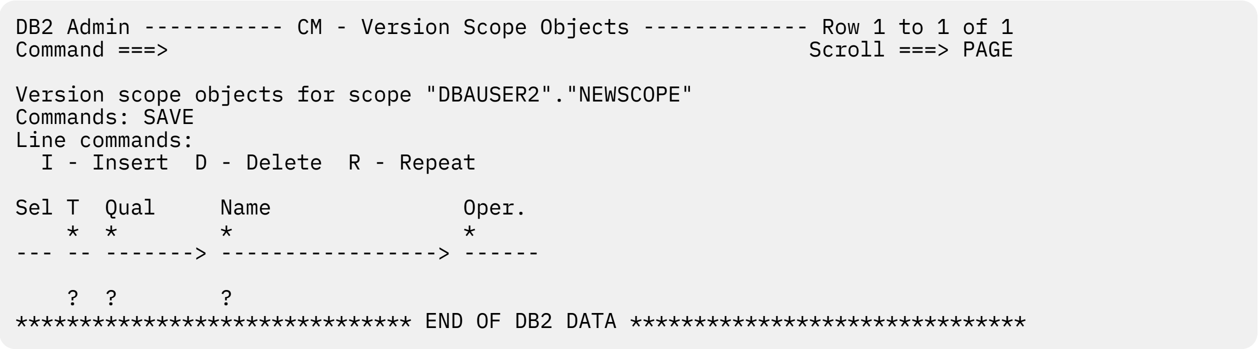
About this task

To create a version scope:

Procedure

1. Specify option 4 on the **Change Management (CM)** panel to display the **Manage Versions** panel.
2. Specify option 3 on the **Manage Versions** panel to display the **Insert Version Scope** panel.
3. Specify a name and owner for the version scope, and, optionally, enter a comment for the version scope.
4. Press F3 to return to the **Manage Versions** panel.
5. Specify option 2 to display the **Version Scopes** panel.
6. Specify the SO line command for the version scope that you created.

   The **Version Scope Objects** panel is displayed, as shown in the following figure:



```
DB2 Admin ----------- CM - Version Scope Objects ------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Version scope objects for scope "DBAUSER2"."NEWSCOPE"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat

Sel T  Qual     Name               Oper.
    *  *        *                  *
--- -- -------> -----------------> ------

    ?  ?        ?
******************************* END OF DB2 DATA *******************************
```

Figure 412. Version Scope Objects panel (ADB2C4O)

7. Use the I line command to add each object that you want in the version scope, and specify the type of object, a qualifier, and a name for the object.

   You can also use the D line command to delete objects from the scope definition, and you can use the R line command to repeat a line to make it faster to define the objects in the scope.

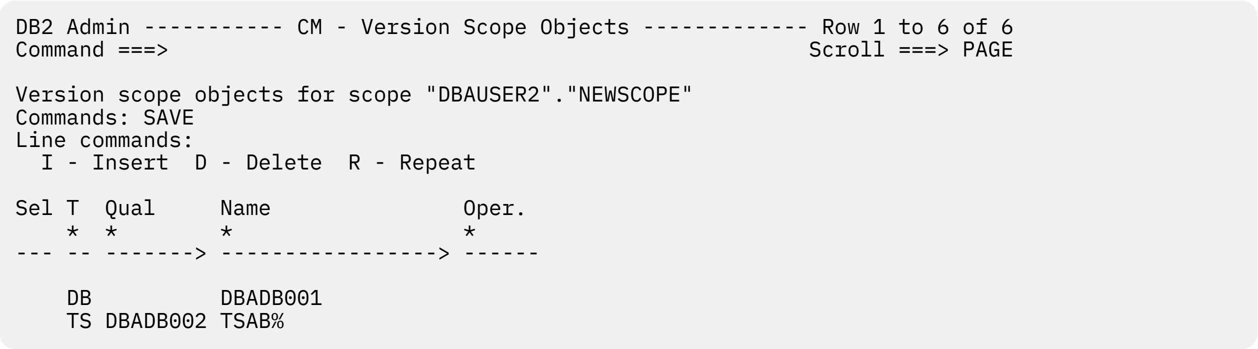
   The values for the qualifier and name can contain zero or more of the following wildcard characters:

   - Minus sign (-) represents any single character.
   - Percent sign (%) or asterisk (*) represents one or more characters.
   - Any other character represents a single occurrence of itself.

   The rules for the wildcard characters follow the rules that are used for the LIKE predicate.

   See the online help for the **Version Scope Objects** panel for a description of the input fields, which includes a list of the types of objects that you can add.

   The following figure shows an example of a version scope definition.



```
DB2 Admin ----------- CM - Version Scope Objects ------------- Row 1 to 6 of 6
Command ===>                                                  Scroll ===> PAGE

Version scope objects for scope "DBAUSER2"."NEWSCOPE"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat

Sel T  Qual     Name               Oper.
    *  *        *                  *
--- -- -------> -----------------> ------

    DB          DBADB001
    TS DBADB002 TSAB%
```

Figure 413. Example of a version scope definition

8. Issue the SAVE primary command to save the definition of the scope.

Deleting version scopes

You can delete a version scope.

About this task

To delete a version scope:

Procedure

1. Specify option 4 on the **Change Management (CM)** panel to display the **Manage Versions** panel.
2. Specify option 2 to display the **Version Scopes** panel.
3. Issue the DEL line command for the version scope that you want to delete.

Displaying version scopes

You can display the version scopes that are stored in the Change Management database.

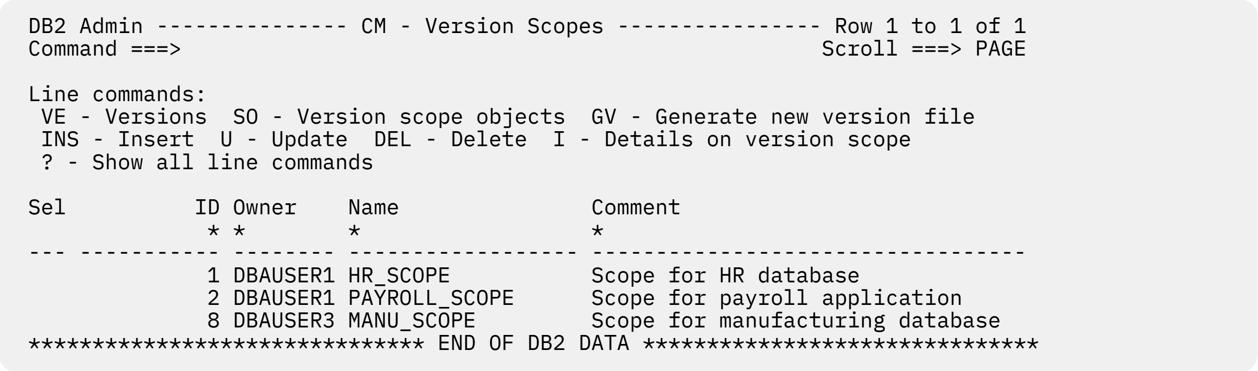
About this task

To display the version scopes:

Procedure

1. Specify option 4 on the **Change Management (CM)** panel to display the **Manage Versions** panel.
2. Specify option 2 to display the **Version Scopes** panel.

   The following figure shows an example of the **Version Scopes** panel:



```
DB2 Admin --------------- CM - Version Scopes ---------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Line commands:
 VE - Versions  SO - Version scope objects  GV - Generate new version file
 INS - Insert  U - Update  DEL - Delete  I - Details on version scope
 ? - Show all line commands

Sel          ID Owner    Name               Comment
              * *        *                  *
--- ----------- -------- ------------------ ----------------------------------
              1 DBAUSER1 HR_SCOPE           Scope for HR database
              2 DBAUSER1 PAYROLL_SCOPE      Scope for payroll application
              8 DBAUSER3 MANU_SCOPE         Scope for manufacturing database
******************************* END OF DB2 DATA *******************************
```

Figure 414. Versions Scopes panel (ADB2C42)

Results

You can issue a variety of line commands on the Version Scopes panel for each version scope. Commands are available to:

- See which versions use the scope
- See which objects are in the scope
- Generate a new base version for the scope
- Insert, delete, or update a scope
- View details about who created the scope and when and who altered it last

Editing version scopes

You can add or delete objects from an existing scope.

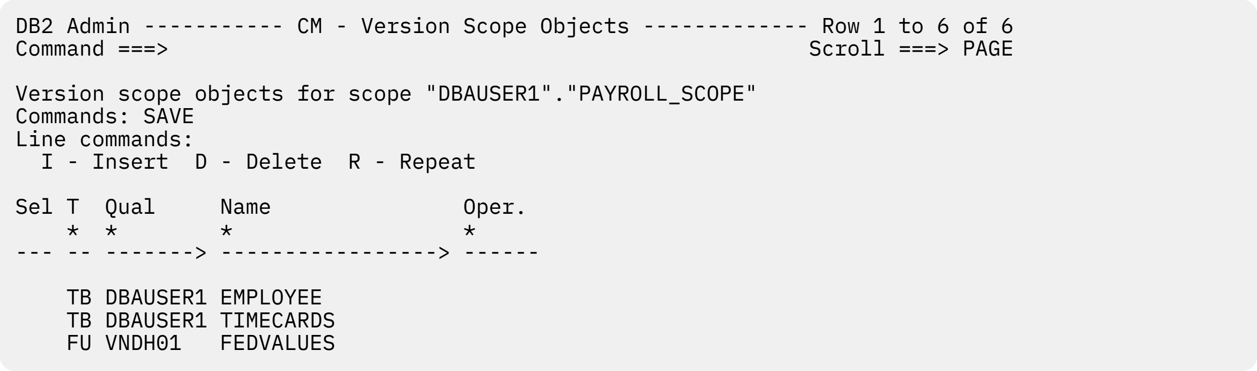
About this task

To edit a version scope:

Procedure

1. Specify option 4 on the **Change Management (CM)** panel to display the **Manage Versions** panel.
2. Specify option 2 to display the **Version Scopes** panel.
3. Specify the SO line command for the version scope that you want to edit.

   The **Version Scope Object** panels, which shows the objects in the current definition, is displayed:



```
DB2 Admin ----------- CM - Version Scope Objects ------------- Row 1 to 6 of 6
Command ===>                                                  Scroll ===> PAGE

Version scope objects for scope "DBAUSER1"."PAYROLL_SCOPE"
Commands: SAVE
Line commands:
  I - Insert  D - Delete  R - Repeat

Sel T  Qual     Name               Oper.
    *  *        *                  *
--- -- -------> ----------------> ------

    TB DBAUSER1 EMPLOYEE
    TB DBAUSER1 TIMECARDS
    FU VNDH01   FEDVALUES
```

Figure 415. Example of editing a version scope definition

4. Use the I and D line commands to insert or delete an object in the definition.

   Ensure that a type, a qualifier, and a name are specified for each object. You can also use the R line command to repeat a line to make it faster to define the objects in the scope.

   See the online help for the Version Scope Objects panel for a description of the input fields, which includes a list of the types of objects that you can add.
5. Issue the SAVE primary command to save the definition of the scope.

Importing a version file

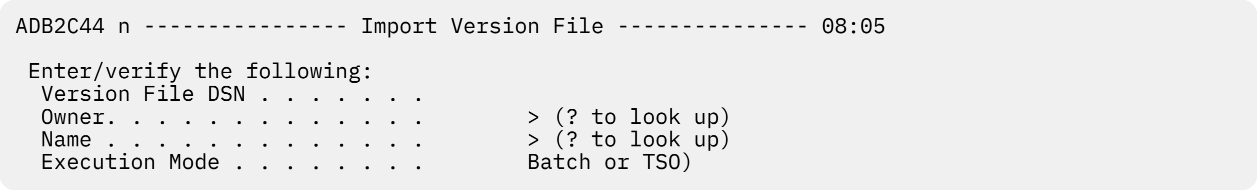
You can import a version file to the change management database.

About this task

To import a version scope:

Procedure

1. Specify option 4 on the **Change Management (CM)** panel to display the **Manage Versions** panel.
2. Specify option 4 to display the **Import Version File** panel.
3. You can specify the following options on the panel:
   - **Version File DSN**: The data set name in which the version file to be imported is contained. The data set can be a stand-alone data set or a PDS with a member
   - **Owner**: The owner of the version to be added to the change management database
   - **Name**: The name of the version to be added to the change management database.
   - **Execution Mode**: Determines whether to import the version in the foreground (TSO) or in the background (batch).



```
ADB2C44 n ---------------- Import Version File --------------- 08:05

 Enter/verify the following:
  Version File DSN . . . . . . .
  Owner. . . . . . . . . . . . .        > (? to look up)
  Name . . . . . . . . . . . . .        > (? to look up)
  Execution Mode . . . . . . . .        Batch or TSO)
```

Figure 416. Example of importing a version file

Quick scopes

Use a quick scope to identify the specific objects to compare in Change Management (CM) batch.

The concept of a quick scope is similar to that of a request parameter for a GEN operation. Request parameters are used to name specific Db2 objects for which SQL is to be generated. Similarly, a quick scope identifies objects to compare in CM batch.

A quick scope has the same syntax and keywords as a request parameter. It also supports the same types as request parameters. For more information about the request parameter syntax and types, see "GEN batch jobs" on page 367.

In addition to the types supported for request parameters, a quick scope also supports the following type:

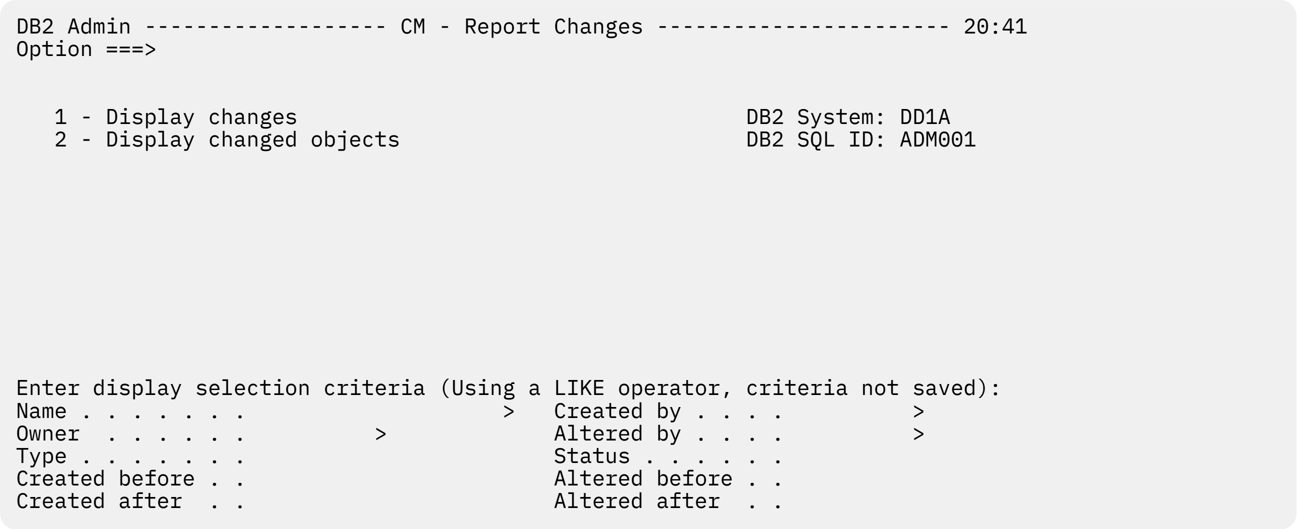
Table 41. The keyword values for quick scope

Object Type	TYPE	QUAL	NAME	Notes®
Db2 Admin Tool version scope	VSCOPE	owner	*name*	VSCOPE is valid only when used to specify a quick scope for the compare source or target in CM batch.

Tracking changes and changed objects

You can use the reporting feature in Change Management to display changes and changed objects and to check the history of changes.

You can use either the **Changes** panel or the **Report Changes** panel to display changes. The **Report Changes** panel, as shown in the following figure, is the main panel for displaying changed objects.



```
DB2 Admin ------------------- CM - Report Changes ----------------------- 20:41
Option ===>


   1 - Display changes                                 DB2 System: DD1A
   2 - Display changed objects                         DB2 SQL ID: ADM001











Enter display selection criteria (Using a LIKE operator, criteria not saved):
Name . . . . . . .                   >   Created by . . . .          >
Owner  . . . . . .          >            Altered by . . . .          >
Type . . . . . . .                       Status . . . . . .
Created before . .                       Altered before . .
Created after  . .                       Altered after  . .
```

Figure 417. Report Changes panel (ADB2C6)

Displaying changes

You can view a list of changes that are stored in the Change Management database. From this list, you can view information about each change and perform actions, such as running or analyzing a change.

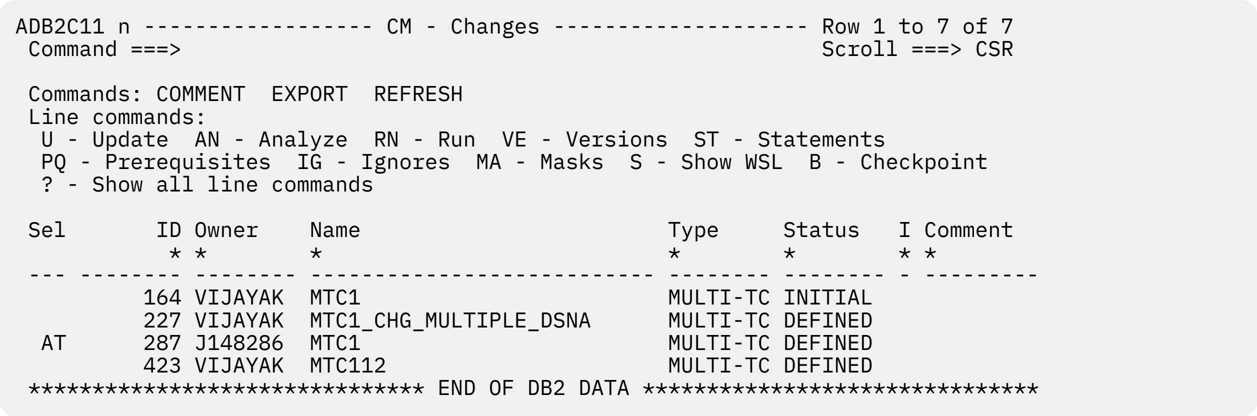
Procedure

To display changes:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option CM, and press Enter.
2. On the **Change Management (CM) (ADB2C)** panel, specify option 1, and press Enter.
3. Optional: At the bottom of the **Manage Changes (ADB2C1)** panel, specify any search criteria to filter or limit the changes that are displayed.

   For example, you can search for changes by owner or status or for changes that were created or altered before or after a certain date. If you want to display all the changes that need to be analyzed, specify DEFINED in the **Status** field.
4. On the **Manage Changes (ADB2C1)** panel, specify option 1, and press Enter.

The **CM - Changes (ADB2C11)** panel displays a list of the changes:



```
ADB2C11 n ------------------ CM - Changes ------------------- Row 1 to 7 of 7
 Command ===>                                                  Scroll ===> CSR

 Commands: COMMENT  EXPORT  REFRESH
 Line commands:
  U - Update  AN - Analyze  RN - Run  VE - Versions  ST - Statements
  PQ - Prerequisites  IG - Ignores  MA - Masks  S - Show WSL  B - Checkpoint
  ? - Show all line commands

 Sel       ID Owner    Name                         Type     Status   I Comment
            * *        *                           *        *        * *
 --- -------- -------- --------------------------- -------- -------- - ---------
          164 VIJAYAK  MTC1                         MULTI-TC INITIAL
          227 VIJAYAK  MTC1_CHG_MULTIPLE_DSNA       MULTI-TC DEFINED
  AT      287 J148286  MTC1                         MULTI-TC DEFINED
          423 VIJAYAK  MTC112                       MULTI-TC DEFINED
 ******************************* END OF DB2 DATA *******************************
```

Tip: When viewing this panel, you can ensure that the latest status of each change is displayed by clicking the REFRESH command.

What to do next

From the **CM - Changes (ADB2C11)** panel, you can use line commands to take actions for each change, such as view the statements for a change (ST), analyze a change (AN), or run a change (RN). Enter the ? line command to see a list of all of the available line commands.

Displaying changed objects

You can display a list of objects that have changes.

About this task

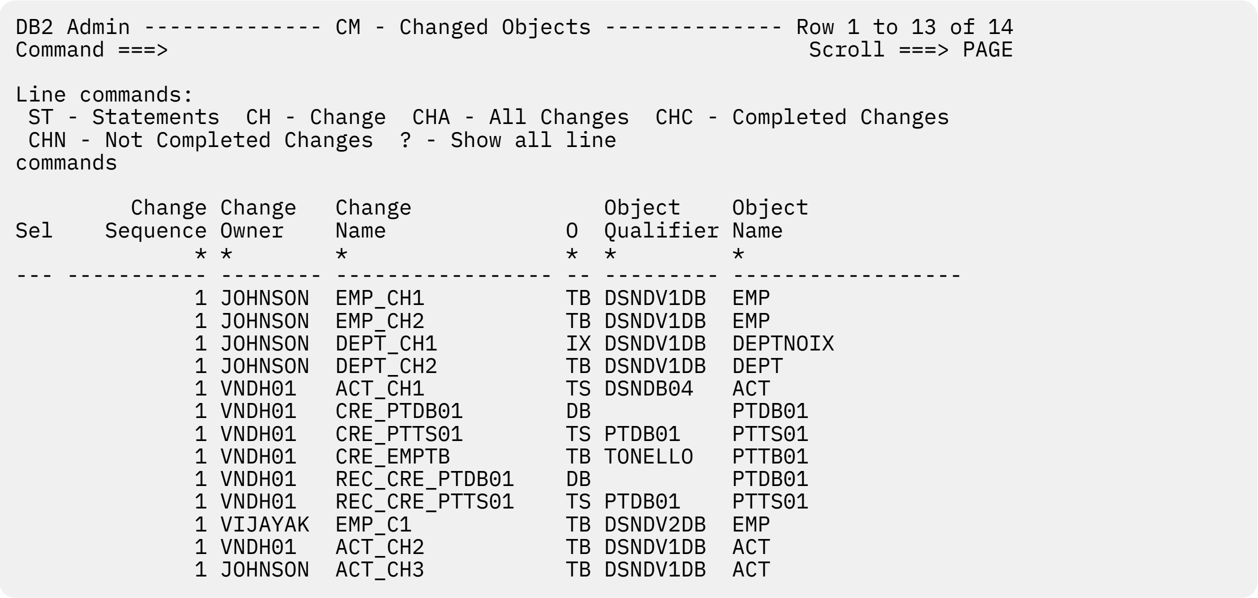
The changes can be in any status and might not be complete.

To display changed objects:

Procedure

1. Specify option 6 on the **Change Management (CM)** panel to display the **Report Changes** panel.
2. Optional: Use the fields at the bottom of the panel to specify the search criteria to filter or limit the objects that are displayed.
   For example, you can specify TB in the **Type** field to display only the tables that have changes. See the online help for a description of the search fields.
3. Specify option 2 to display the **Changed Objects** panel.

   The following figure shows an example of the **Changed Objects** panel:



```
DB2 Admin -------------- CM - Changed Objects -------------- Row 1 to 13 of 14
Command ===>                                                  Scroll ===> PAGE

Line commands:
 ST - Statements  CH - Change  CHA - All Changes  CHC - Completed Changes
 CHN - Not Completed Changes  ? - Show all line
commands

         Change Change   Change              Object    Object
Sel    Sequence Owner    Name              O  Qualifier Name
              * *        *                 *  *         *
--- ----------- -------- ----------------- -- --------- ------------------
              1 JOHNSON  EMP_CH1           TB DSNDV1DB  EMP
              1 JOHNSON  EMP_CH2           TB DSNDV1DB  EMP
              1 JOHNSON  DEPT_CH1          IX DSNDV1DB  DEPTNOIX
              1 JOHNSON  DEPT_CH2          TB DSNDV1DB  DEPT
              1 VNDH01   ACT_CH1           TS DSNDB04   ACT
              1 VNDH01   CRE_PTDB01        DB           PTDB01
              1 VNDH01   CRE_PTTS01        TS PTDB01    PTTS01
              1 VNDH01   CRE_EMPTB         TB TONELLO   PTTB01
              1 VNDH01   REC_CRE_PTDB01    DB           PTDB01
              1 VNDH01   REC_CRE_PTTS01    TS PTDB01    PTTS01
              1 VIJAYAK  EMP_C1            TB DSNDV2DB  EMP
              1 VNDH01   ACT_CH2           TB DSNDV1DB  ACT
              1 JOHNSON  ACT_CH3           TB DSNDV1DB  ACT
```

Figure 418. Changed Objects panel (ADB2C62)

4. Optional: Use the line commands to perform various actions on a changed object.
   For example, you can display all the completed changes for a particular object or you can get details on a particular change.

REST services

You can use Db2 Admin Tool to create and manage your own REST services. Additionally, Db2 Admin Tool provides some REST APIs that you can use to call certain features of Db2 Admin Tool.

Managing Db2 REST services

You can use Db2 as a REST service provider. Db2 defines a REST service as a package. Each package contains a single static SQL statement, and information about each REST service package is stored in a user-defined table, SYSIBM.DSNSERVICE.

Before you begin

You must be authorized to access SYSIBM.DSNSERVICE. You can use the following statement to grant access:

```
GRANT SELECT ON TABLE SYSIBM.DSNSERVICE TO PUBLIC;
```

PUBLIC is site-specific.

About this task

You can use Db2 Admin Tool to display information about Db2 REST services, start and stop them, and free the corresponding package for a REST service.

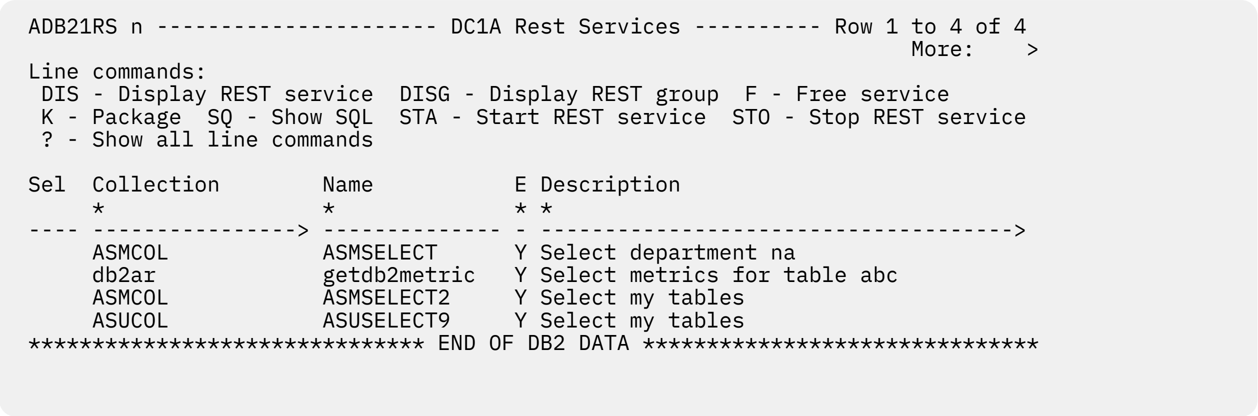
For instructions on how to create a REST service in Db2 Admin Tool, see “Creating a Db2 REST service” on page 873.

Procedure

To manage Db2 REST services:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option RS, and press Enter.

If no REST services exist, the **Create Rest Service (ADB21RSB)** panel is displayed so that you can create one. Otherwise, the **REST Services (ADB21RS)** panel is displayed with information about each REST service from SYSIBM.DSNSERVICE:



```
ADB21RS n --------------------- DC1A Rest Services ---------- Row 1 to 4 of 4
                                                                    More:    >
Line commands:
 DIS - Display REST service  DISG - Display REST group  F - Free service
 K - Package  SQ - Show SQL  STA - Start REST service  STO - Stop REST service
 ? - Show all line commands

Sel  Collection        Name           E Description
     *                 *              * *
---- ----------------> ------------- - ------------------------------------>
     ASMCOL            ASMSELECT      Y Select department na
     db2ar             getdb2metric   Y Select metrics for table abc
     ASMCOL            ASMSELECT2     Y Select my tables
     ASUCOL            ASUSELECT9     Y Select my tables
****************************** END OF DB2 DATA ******************************
```

For a detailed description of the panel columns and line commands, see “Option RS. REST Services” on page 172.

3. Take one or more of the following actions as needed:

   - **To display the status of a REST service:** Use the DIS or DISG line command. (DISG displays the status on all members of the data sharing group.) Db2 Admin Tool issues the Db2 command DIS RESTSVC.
   - **To display information about the package that contains the rest service:** Use the K line command.
   - **To view the SQL statement in the REST service package:** Use the SQ line command.
   - **To start a service:** Use the STA or STAG line command. (STAG starts the service on all members of the data sharing group.) Db2 Admin Tool issues the Db2 command STA RESTSVC.
   - **To stop a service:** Use the STO or STOG line command. (STOG stops the service on all members of the data sharing group.) Db2 Admin Tool issues the Db2 command STO RESTSVC.
   - **To free the package that contains the REST service:** Use the F line command. After you free the package and return to this panel, the service is no longer listed.

Related tasks

“Creating a Db2 REST service” on page 873
You can create a new Db2 REST service in Db2 Admin Tool if you have the appropriate authority.

Related reference

“Option RS. REST Services” on page 172
The **REST Services (ADB21RS)** panel displays information about Db2 REST services.

Related information

Db2 REST services (Db2 12 for z/OS)

Video: Db2 Administration Tool: Db2 REST services support

Creating a Db2 REST service

You can create a new Db2 REST service in Db2 Admin Tool if you have the appropriate authority.

Before you begin

- Db2 REST services must be enabled. For instructions, see Enabling Db2 REST services (Db2 12 for z/OS).
- You must have the required authority to create a service and bind the package. For more information, see Creating a Db2 REST service (Db2 12 for z/OS).

- You must be authorized to access SYSIBM.DSNSERVICE. You can use the following statement to grant access:

  ```
  GRANT SELECT ON TABLE SYSIBM.DSNSERVICE TO PUBLIC;
  ```

  PUBLIC is site-specific.

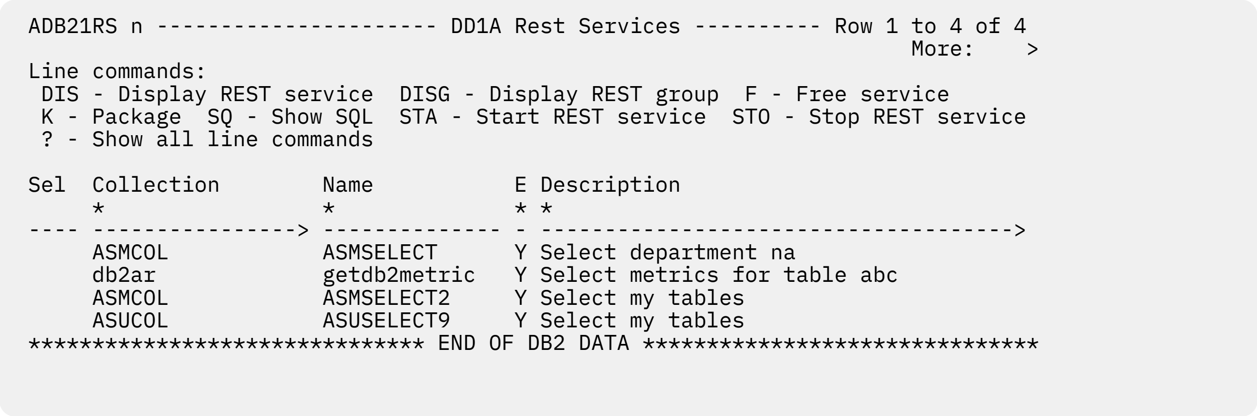
Procedure

To create a Db2 REST service:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option RS, and press Enter.

   If no REST services exist, the **Create Rest Service (ADB21RSB)** panel is displayed. Skip to step "4" on page 874.

   If REST services exist, the **REST Services (ADB21RS)** panel is displayed with information about any existing REST services:

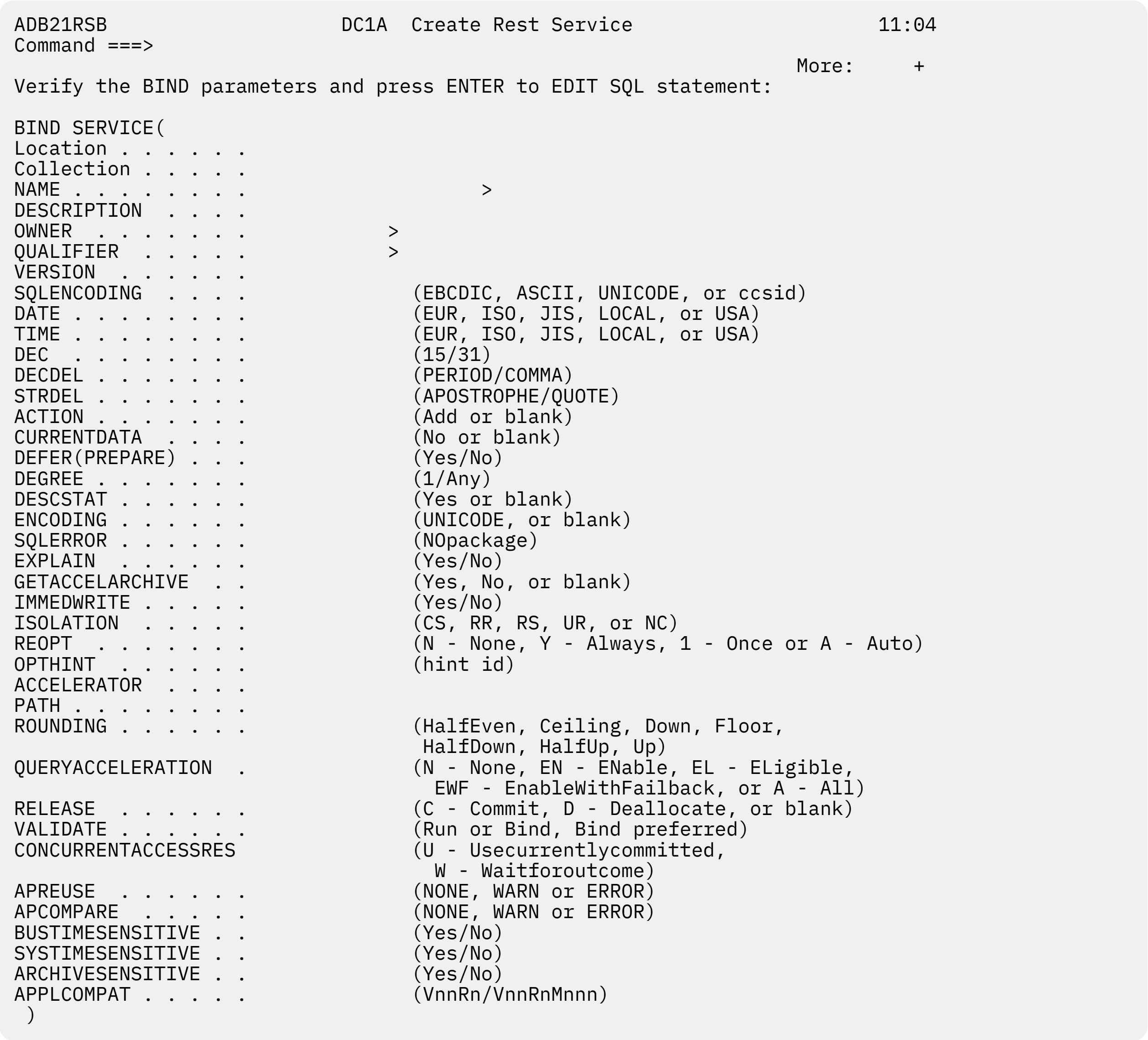


```
ADB21RS n ---------------------- DD1A Rest Services ---------- Row 1 to 4 of 4
                                                                     More:    >
Line commands:
 DIS - Display REST service  DISG - Display REST group  F - Free service
 K - Package  SQ - Show SQL  STA - Start REST service  STO - Stop REST service
 ? - Show all line commands

Sel  Collection        Name           E Description
     *                 *              * *
---- ----------------> -------------- - ------------------------------------->
     ASMCOL            ASMSELECT      Y Select department na
     db2ar             getdb2metric   Y Select metrics for table abc
     ASMCOL            ASMSELECT2     Y Select my tables
     ASUCOL            ASUSELECT9     Y Select my tables
******************************* END OF DB2 DATA ********************************
```

   *Figure 419. **REST Services (ADB21RS)** panel*

   Tip: If the name, collection, or version of your rest service will have lowercase characters or special characters, specify the CAPS OFF primary command on this panel before preceding to the next step.

3. Specify the CRE or B line command, and press Enter.
4. On the **Bind Rest Service (ADB21RSB)** panel or **Create Rest Service (ADB21RSB)** panel, specify the appropriate BIND options and press Enter:



```
ADB21RSB                    DC1A  Create Rest Service                     11:04
Command ===>
                                                                   More:     +
Verify the BIND parameters and press ENTER to EDIT SQL statement:

BIND SERVICE(
Location . . . . . .
Collection . . . . .
NAME . . . . . . . .                    >
DESCRIPTION  . . . .
OWNER  . . . . . . .            >
QUALIFIER  . . . . .            >
VERSION  . . . . . .
SQLENCODING  . . . .              (EBCDIC, ASCII, UNICODE, or ccsid)
DATE . . . . . . . .              (EUR, ISO, JIS, LOCAL, or USA)
TIME . . . . . . . .              (EUR, ISO, JIS, LOCAL, or USA)
DEC  . . . . . . . .              (15/31)
DECDEL . . . . . . .              (PERIOD/COMMA)
STRDEL . . . . . . .              (APOSTROPHE/QUOTE)
ACTION . . . . . . .              (Add or blank)
CURRENTDATA  . . . .              (No or blank)
DEFER(PREPARE) . . .              (Yes/No)
DEGREE . . . . . . .              (1/Any)
DESCSTAT . . . . . .              (Yes or blank)
ENCODING . . . . . .              (UNICODE, or blank)
SQLERROR . . . . . .              (NOpackage)
EXPLAIN  . . . . . .              (Yes/No)
GETACCELARCHIVE  . .              (Yes, No, or blank)
IMMEDWRITE . . . . .              (Yes/No)
ISOLATION  . . . . .              (CS, RR, RS, UR, or NC)
REOPT  . . . . . . .              (N - None, Y - Always, 1 - Once or A - Auto)
OPTHINT  . . . . . .              (hint id)
ACCELERATOR  . . . .
PATH . . . . . . . .
ROUNDING . . . . . .              (HalfEven, Ceiling, Down, Floor,
                                   HalfDown, HalfUp, Up)
QUERYACCELERATION  .              (N - None, EN - ENable, EL - ELigible,
                                    EWF - EnableWithFailback, or A - All)
RELEASE  . . . . . .              (C - Commit, D - Deallocate, or blank)
VALIDATE . . . . . .              (Run or Bind, Bind preferred)
CONCURRENTACCESSRES               (U - Usecurrentlycommitted,
                                    W - Waitforoutcome)
APREUSE  . . . . . .              (NONE, WARN or ERROR)
APCOMPARE  . . . . .              (NONE, WARN or ERROR)
BUSTIMESENSITIVE . .              (Yes/No)
SYSTIMESENSITIVE . .              (Yes/No)
ARCHIVESENSITIVE . .              (Yes/No)
APPLCOMPAT . . . . .              (VnnRn/VnnRnMnnn)
 )
```

*Figure 420. **Create Rest Service (ADB21RSB)** panel*

Note: The title of panel ADB21RSB can be either **Bind Rest Service** or **Create Rest Service**, depending on the line command that you used to display this panel.

5. On the **Edit Statement (ADB2PRED)** panel, enter the SQL statement that you want in the service, and press End (PF3).
6. After you create the service, issue the REF command on the **REST Services (ADB21RS)** panel to refresh the list and see your new service.

Related tasks

“Managing Db2 REST services” on page 872
You can use Db2 as a REST service provider. Db2 defines a REST service as a package. Each package contains a single static SQL statement, and information about each REST service package is stored in a user-defined table, SYSIBM.DSNSERVICE.

Related reference

“Option RS. REST Services” on page 172
The **REST Services (ADB21RS)** panel displays information about Db2 REST services.

Related information

Db2 REST services (Db2 12 for z/OS)

Video: Db2 Administration Tool: Db2 REST services support

Provided REST APIs

Db2 Admin Tool provides REST APIs for the following functions: space manager and DDL generator.

Each of these provided REST APIs is a package that contains a single call to a stored procedure. These stored procedures are provided by Db2 Admin Tool and invoke the relevant feature of the tool. (This implementation conforms to the Db2 definition of a REST service as a package that contains a single static SQL statement.)

The following table lists information about the provided REST APIs, including the corresponding ISPF panel or command that provides the same functionality, if applicable.

Table 42. REST APIs provided by Db2 Admin Tool

Db2 Admin Tool function	REST API [1]	Description	Panel or command that provides the same functionality
Space manager	ADB2ME2	Estimates the number of extents needed	**DB2 Extents Estimator (ADB2ME2)** panel
Space manager	ADB2MES	Provides space estimates for table spaces	**DB2 Table Space Estimator (ADB2MES)** panel
Space manager	ADB2MEX	Provides space estimates for index spaces	**DB2 Index Space Estimator (ADB2MEX)** panel
DDL generator	ADBGDDL	Generates DDL for one object	The DDL line command on an object

Note:

1. The associated stored procedure has the same name as the REST API.

Before you can call one of these REST APIs, you must create the relevant stored procedure and bind it as a REST service. Jobs and instructions are provided; see "Setting up the provided REST APIs" on page 118.

Calling a provided REST API

You can call certain features in Db2 Admin Tool as REST APIs by using one of the provided stored procedures. For example, you can get space estimates for index spaces by using the ADB2MEX stored procedure.

About this task

For a list of available REST APIs and the corresponding stored procedure name, see "Provided REST APIs" on page 876.

Procedure

To call a Db2 Admin Tool function as a REST API:

1. Set up the provided REST APIs.
2. Call one of the provided REST APIs by using an API development tool.

   The process varies based on the tool. Generally, you need to create a new request, specify the URL for your REST service, select the POST method, and provide your input data in JSON format. For example, you can provide the following input data for ADB2MEX:

```
{ "ROWS": 3334455,
"KEYLEN": 2000,
"UNIQUE": "y",
"DISTINCT": null,
"ROWSPKEY": null,
"CMPRATIO": null,
"PAGESIZE": null,
"PCTFREE": null,
"FREEPAGE": null,
"LARGETS": "Y",
"UNITTYPE": null,
"EAV": null,
"PIECENUM": null,
"PIECESIZE": ""
}
```

Also, by default, sufficient authorization is required for the stored procedure. You must pass valid LPAR credentials (login ID and password) to call the procedure.

After you send the request, the relevant stored procedure sends a response. For example, the following response is for ADB2MEX:

```

{
    "Output Parameters":
{      "RES": "\
{\"rc\":0,\"pgusea\":3836,\"pgrows\":1,\"pgleaf\":3334455,\"ixlevels\":22,\"pgtot\":6669743,\
"estkb\":26678972,\"pqty\":26679600,\"pqtyexpl\":\"
(37055 cyls)\",\"sqty\":2668320,\"sqtyexpl\":\"
(3706 cyls)\",\"sugpiece\":\"2097152
K\",\"esttrks\":555812,\"estcyls\":37055,\"snum\":12,\"msg\":\"Successful completion\"}
"
    },
    "StatusCode": 200,
    "StatusDescription": "Execution Successful"
}
```

In the case of ADB2MEX, if an error occurs (for example, some required parameters are missing), the response from the procedure contains the return code and an error message. For example:

```
{
    "Output Parameters":
{      "RES": "\{\"rc\":-1,\"emsg\":\"One or more of the required parameters are missing\"}
"
    },
    "StatusCode": 200,
    "StatusDescription": "Execution Successful"
}
```

ADB2ME2 stored procedure

The ADB2ME2 stored procedure returns the estimated number of extents needed for a table space or index space based on the provided input parameter values. This REXX stored procedure invokes the space estimator feature of Db2 Admin Tool and corresponds to the functionality provided on the **DB2 Extents Estimator (ADB2ME2)** panel.

Environment

The ADB2ME2 stored procedure runs in a WLM-established stored procedure address space. For setup instructions, see "Setting up the provided REST APIs" on page 118.

Authorization

To execute the CALL statement, the owner of the package or plan that contains the CALL statement must have EXECUTE privilege on the ADB2ME2 stored procedure.

Syntax

▶▶─ CALL ── ADB2ME2 ── (── *priqty* , ── *secqty* , ── *estkb*, ── *unittype* , ── *res* ──) ─▶◀

Input parameters

priqty
The primary space allocation in kilobytes (KB).

Type:
BIGINT

secqty
The secondary space allocation in KB.

Type:
BIGINT

estkb
The estimated number of KB required for the table space.

Type:
BIGINT

unittype
The unit type. This value is used when calculating the estimated number of tracks and cylinders.

Type:
VARCHAR(10)

Acceptable values:

3380
3390

Output parameters

res
A string that contains the output data. The string contains the return code, the space estimates, and a completion message, as shown in the following example res string:

```
"rc\":0,\"estext\":2,\"pqinexpl\":\"(1 track)\",\"sqinexpl\":\"(1 track)
\",\"msg\":\"Successful completion\"
```

Within this string, a backslash (\) is used as an escape character for the quotation marks ("). The estimate values in this string correspond to fields on the **DB2 Extents Estimator (ADB2ME2)** panel as follows:

Table 43. res estimate values

Value in *res* string	**Corresponding estimates field on the DB2 Extents Estimator (ADB2ME2) panel**
`estext`	**Estimated extents**
`pqinexpl`	**PRIQTY** (the second value in parentheses, for example `2 cyls`)
`sqinexpl`	**SECQTY** (the second value in parentheses, for example, `1 track`)

Type:
VARCHAR

Examples

The following example CALL statement returns an extent estimate given the specified input parameters:

```
CALL ADB2ME2(2417483647, 2417483647, 214, '3390', :RES);
```

The following example CALL statement in an application program passes host variables as the input parameters and returns an extent estimate based on those variable values at the time the application is run.

```
CALL ADB2ME2(:PRIQTY, :SECQTY, :ESTKB, :UNITTYPE, :RES);
```

Related tasks

“Estimating space requirements for table spaces” on page 635
You can use the space manager feature of Db2 Admin Tool to estimate the space requirements for a table space. For example, if you plan to add a significant number of rows to a table, you might want to first determine how much space this addition requires.

ADB2MES stored procedure

The ADB2MES stored procedure returns the estimated space required for a table space based on the provided input parameter values. This REXX stored procedure invokes the space estimator feature of Db2 Admin Tool and corresponds to the functionality provided on the **DB2 Table Space Estimator (ADB2MES)** panel.

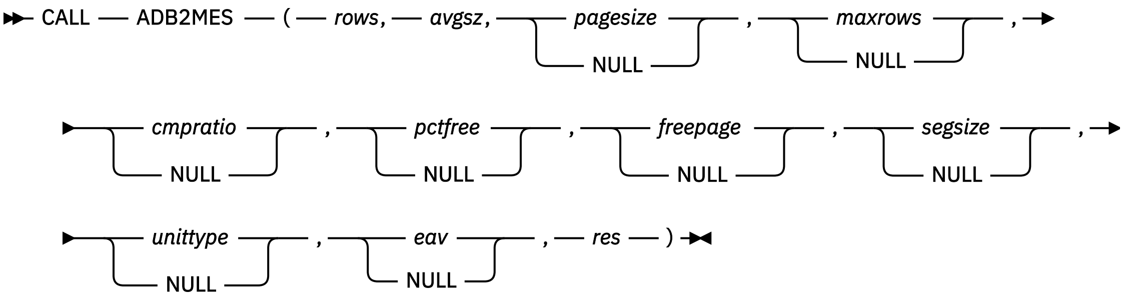
Environment

The ADB2MES stored procedure runs in a WLM-established stored procedure address space. For setup instructions, see “Setting up the provided REST APIs” on page 118.

Authorization

To execute the CALL statement, the owner of the package or plan that contains the CALL statement must have EXECUTE privilege on the ADB2MES stored procedure.

Syntax



Input parameters

rows
The number of rows to be loaded into the table space.

Type:
INTEGER

avgsz
The average record size before data compression, including 1 byte for each field that allows null and 2 bytes for each varying-length field.

Type:
INTEGER

pagesize
The page size in kilobytes (KB).

Type:
INTEGER

Default value:
4

maxrows
The maximum number of rows per page.

Type:
INTEGER

Default value:
255

cmpratio
The compression ratio, which is the percentage of space that is used to hold a compressed row compared to an uncompressed row.

Type:
DOUBLE

Default value:
0

pctfree
The percentage of each page to leave as free space when the table is loaded or reorganized.

Type:
INTEGER

Default value:
5

freepage
The number of pages after which Db2 is to leave a page of free space when the table is loaded or reorganized.

Type:
INTEGER

Default value:
0

segsize
The number of pages to be assigned to each segment.

Type:
INTEGER

Default value:
0

unittype
The unit type. This value is used when calculating the estimated number of tracks and cylinders.

Type:
VARCHAR(10)

Acceptable values:

3380
3390

Default value:
3390

eav
An indication of whether Extended Address Volumes (EAV) are used. If this value is set to YES, the space estimate is increased by 10 cylinders and then rounded up to a multiple of 21 cylinders.

Type:
VARCHAR(1)

Acceptable values:

Y
N

Default value:
N

Output parameters

res

A string that contains the output data. The string contains the return code, the space estimates, and a completion message, as shown in the following example *res* string:

```
"rc\":0,\"pgusea\":7731,\"pgrows\":1,\"pgused\":4,\"pgtot\":22,\"estkb\":176,\"pqty\":192,\"
sqty\":19,\"pqtyexpl\":\"
(4 tracks)\",\"sqtyexpl\":\"(1 track)\",\"esttrks\":4,\"estcyls\":1,\"msg\":"Page size has
been set to 8 according to maximum record size\"
```

Within this string, a backslash (\) is used as an escape character for the quotation marks ("). The estimate values in this string correspond to fields on the **DB2 Table Space Estimator (ADB2MES)** panel as follows:

Table 44. res estimate values

Value in *res* string	**Corresponding estimates field on the DB2 Table Space Estimator (ADB2MES) panel**
`pgusea`	**Usable page size**
`pgrows`	**Rows per page**
`pgused`	**Pages used**
`pgtot`	**Total pages**
`estkb`	**Number of KB**
`pqty`	**Primary** (the first value)
`sqty`	**Secondary** (the first value)
`pqtyexpl`	**Primary** (the second value in parentheses, for example, `2 cyls`)
`sqtyexpl`	**Secondary** (the second value in parentheses, for example, `1 track`)
`esttrks`	**Number of trks**
`estcyls`	**Number of cyls**

Type:
VARCHAR

Examples

The following example CALL statement returns the space estimate for the table space given the specified input parameters:

```
CALL ADB2MES(0, 4048, 4, 255, 99, 5, 0, 4, '3390', 'Y', :RES);
```

The following example CALL statement in an application program passes host variables as the input parameters and returns the space estimate for the table space based on those variable values at the time the application is run.

```
CALL
ADB2MES(:ROWS, :AVGSZ, :PAGESIZE, :MAXROWS, :CMPRATIO, :PCTFREE, :FREEPAGE, :SEGSIZE, :UNITTYPE,
 :EAV, :RES);
```

Related tasks

“Estimating space requirements for table spaces” on page 635
You can use the space manager feature of Db2 Admin Tool to estimate the space requirements for a table space. For example, if you plan to add a significant number of rows to a table, you might want to first determine how much space this addition requires.

ADB2MEX stored procedure

The ADB2MEX stored procedure returns the estimated space required for index spaces based on the provided input parameter values. This REXX stored procedure invokes the space estimator feature of Db2 Admin Tool and corresponds to the functionality provided on the **DB2 Index Space Estimator (ADB2MEX)** panel.

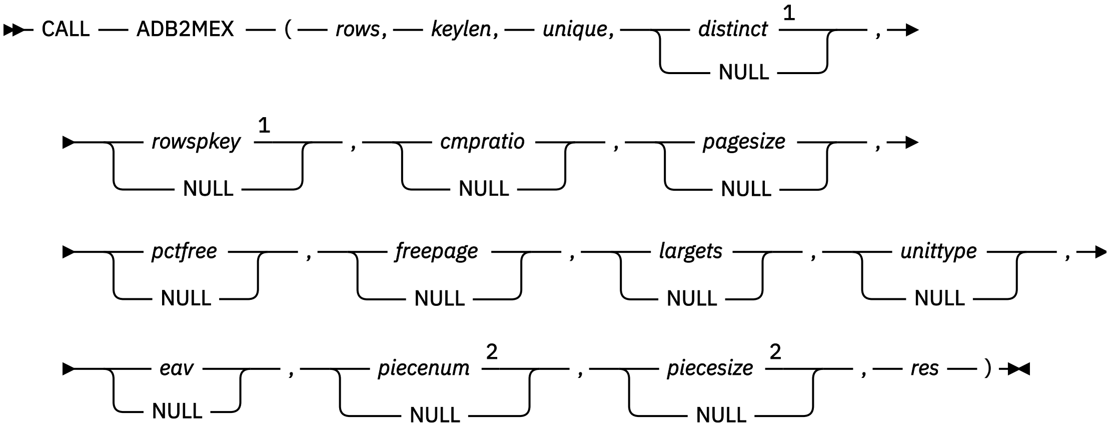
Environment

The ADB2MEX stored procedure runs in a WLM-established stored procedure address space. For setup instructions, see “Setting up the provided REST APIs” on page 118.

Authorization

To execute the CALL statement, the owner of the package or plan that contains the CALL statement must have EXECUTE privilege on the ADB2MEX stored procedure.

Syntax



Notes:

[1] *distinct* or *rowspkey* is required only if *unique* is N. You can specify a value for *distinct* or *rowspkey* but not both.
[2] You can specify a value for *piecenum* or *piecesize* but not both.

Input parameters

rows
The number of keys in the index that refer to data rows.

Type:
INTEGER

keylen
The sum of the length of all the columns of the key, plus the number of the columns that allow nulls.

Type:
INTEGER

unique
An indication of whether the key is unique.

Type:
CHAR(1)

Acceptable values:

Y
N

distinct
The number of distinct keys.

Type:
INTEGER

rowspkey
The average number of rows per distinct key.

Type:
INTEGER

cmpratio
The compression ratio, which is the percentage of space that is used after compression compared to space used without compression.

Type:
DOUBLE

Default value:
0

pagesize
The page size in KB.

Type:
INTEGER

Default value:
4

pctfree
The percentage of each page to leave as free space when the table is loaded or reorganized.

Type:
INTEGER

Default value:
5

freepage
The number of pages after which Db2 is to leave a page of free space when the table is loaded or reorganized.

Type:
INTEGER

Default value:
0

largets
An indication of whether the table space for this index is defined as LARGE.

Type:
CHAR(1)

Acceptable values:

Y
N

Default value:
N

unittype

The unit type. This value is used when calculating the estimated number of tracks and cylinders.

Type:
VARCHAR(10)

Acceptable values:

```
3380
3390
```

Default value:
```
3390
```

eav

An indication of whether Extended Address Volumes (EAV) are used. If this value is set to Y, the space estimate is increased by 10 cylinders and then rounded up to a multiple of 21 cylinders.

Type:
VARCHAR(1)

Acceptable values:

Y
N

Default value:
N

piecenum

The number of data set pieces into which the index is to be split.

Type:
INTEGER

piecesize

The piece size in kilobytes (K), megabytes (M), or gigabytes (G).

Type:
VARCHAR(12)

Acceptable values:

For K: 256,512,1024,2048,4096,8192, and so on up to 2^{28}
For M: 1,2,4,8,16,32,64,128,256,512,1024, and so on up to 2^{18}
For G: 1,2,4,8,16,32,64,128, and 256

Examples

```
1024M
1G
4096K
```

Output parameters

res

A string that contains the output data. The string contains the return code, the space estimates, and a completion message, as shown in the following example res string:

```
"rc\":0,\"pgusea\":3836,\"pgrows\":1,\"pgleaf\":3334455,\"ixlevels\":22,\"pgtot\":6669743,\"
estkb\":26678972,\"pqty\":26679600,\"pqtyexpl\":\"
(37055 cyls)\",\"sqty\":2668320,\"sqtyexpl\":\"
(3706 cyls)\",\"sugpiece\":\"2097152
K\",\"esttrks\":555812,\"estcyls\":37055,\"snum\":12,\"msg\":\"Successful completion\"
```

Within this string, a backslash (\) is used as an escape character for the quotation marks ("). The estimate values in this string correspond to fields on the **DB2 Index Space Estimator (ADB2MEX)** panel as follows:

Table 45. res estimate values

Value in *res* string	**Corresponding estimates field on the DB2 Index Space Estimator (ADB2MEX) panel**
`pgusea`	**Usable page size**
`pgrows`	**Keys per page**
`pgleaf`	**Leaf pages**
`ixlevels`	**Index levels**
`pgtot`	**Total pages**
`estkb`	**Number of KB**
`pqty`	**Primary** (the first value)
`pqtyexpl`	**Primary** (the second value in parentheses, for example, `2 cyls`)
`sqty`	**Secondary** (the first value)
`sqtyexpl`	**Secondary** (the second value in parentheses, for example, `1 track`)
`sugpiece`	**Piecesize**
`esttrks`	**Number of trks**
`estcyls`	**Number of cyls**
`snum`	**Piecesize** (the second value in parentheses)

Type:
VARCHAR

Examples

The following example CALL statement returns the space estimate for the index space given the specified input parameters:

```
CALL ADB2MEX(350, 156, 'Y', NULL, NULL, 0, 4, 10, 0, 'N', 3390, 'N', NULL, '4G', :RES);
```

The following example CALL statement in an application program passes host variables as the input parameters and returns the space estimate for the index space based on those variable values at the time the application is run.

```
CALL ADB2MEX(:ROWS, :KEYLEN, :UNIQUE, :DISTINCT, :ROWSPKEY, :CMPRATIO, :PAGESIZE,
:PCTFREE, :FREEPAGE, :LARGETS, :UNITTYPE, :EAV, :PIECENUM, :PIECESIZE :RES);
```

Related tasks

"Estimating space requirements for index spaces" on page 636

You can use the space manager feature of Db2 Admin Tool to estimate the space requirements for an index space. For example, if you plan to add a significant amount of data to a table, you might want to determine how this additional will impact the space that is needed for an index.

ADBGDDL stored procedure

Use the ADBGDDL stored procedure to get the DDL for a single object from the Db2 catalog.

The input parameters identify the object for which you want the DDL. The output parameters provide information about whether the operation to generate the DDL was successful. The core function that ADBGDDL uses to generate the DDL is the GEN function.

ADBGDDL returns DDL in a result set and optionally returns a second result set with the GEN report.

Environment

The ADBGDDL stored procedure runs in a WLM-established stored procedure address space. For setup instructions, see "Setting up the provided REST APIs" on page 118.

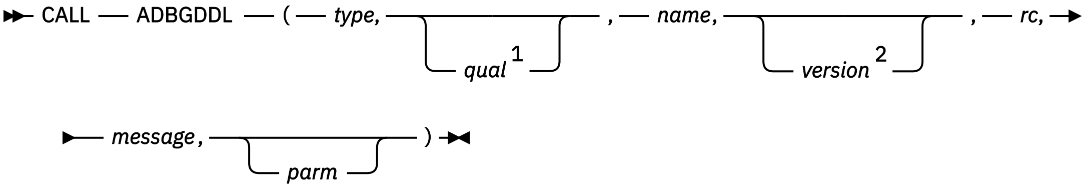
Recommendation: IBM strongly recommends that ADBGDDL only is configured to run in the same WLM environment as the Db2-supplied stored procedure ADMIN_INFO_SYSPARM if the WLM environment is configured with MNSPAS>1. (MNSPAS is the parameter for the number of address spaces.)

ADBGDDL calls ADMIN_INFO_SYSPARM to retrieve Db2 subsystem parameter information. ADMIN_INFO_SYSPARM must run in a WLM environment with NUMTCB=1. Therefore, if these two procedures run in the same WLM environment with NUMTCB=1 and MNSPAS=1, ADBGDDL will wait for the call to ADMIN_INFO_SYSPARM to fail, because no WLM address space is available. ADBGDDL will continue after this failure but not be able to optimize the DDL based on the subsystem parameters. Also, the response time will be significantly higher because of the wait for ADMIN_INFO_SYSPARM.

Authorization

To execute the CALL statement, the owner of the package or plan that contains the CALL statement must have EXECUTE privilege on the ADBGDDL stored procedure.

Syntax



Notes:

[1] *qual* is required for objects that can be qualified. Otherwise, this parameter is invalid.
[2] *version* is optional for functions, procedures, and triggers. Otherwise, this parameter is invalid.

Input parameters

type

The object type. You can specify one of the following values:

AL
Alias

DB
Database

DT
Data type

FU
Function

GV
Global variable

IX
Index

MK
Column mask

PM
Permission

QA
Sequence alias

RO
Role

SG
Storage group

SP
Stored procedure

SQ
Sequence

SY
Synonym

TB
Table

TC
Trusted context

TG
Trigger

TS
Table space

VW
View

qual
The object qualifier.

name
The object name.

version
The version of the object.

parm
An optional parameter. Currently, you can specify only the following parameter:

RPT=Y
Indicates that the GEN report is to be returned. The default is to not produce a report.

Output parameters

rc
The return code.

Type:
SMALLINT

message
A message that describes the result of the operation, such as a formatted SQL message. This message can be a success or error message.

Type:
VARCHAR(800)

Result sets

Result set	Description
C_SQL	The generated DDL. If an error occurred, this result set is empty. The format of this result set is the same as the SQL result set for ADB2RE. For a description of the columns in this result set, see "SQL result set" on page 385.
C_RPT	An optional result set to contain the GEN REPORT. This result set is returned if the parameter RPT=Y was specified. The format of this result set is the same as the report result set for ADB2RE. For a description of the columns in this result set, see "Report result set" on page 385.

Example

Issue the following CALL statement to generate a CREATE DATABASE statement for the Db2 catalog database DSNDB06 and return result sets for the SQL and the report, a return code, and a message:

```
CALL ADBGDDL('DB',
             '',
             'DSNDB06',
             '',
             :rc,
             :msg,
             'RPT=Y');
```

Determining whether applications need to be rebound

Bind avoidance is a feature of Db2 Admin Tool that helps you avoid unnecessary bind operations. Unnecessary binds can waste resources and potentially cause a Db2 access path change that might negatively affect performance.

The bind avoidance feature analyzes code changes in your application to determine whether a bind operation is needed. For example, you need to rebind if you change the SQL statements in your application. You also need to rebind if you make changes that affect the plan or package, such as creating an index, even if you have not changed the SQL statements. You do not need to rebind for source code changes that do not alter the existing SQL structure. For more information about situations that require bind or rebind operations, see Rebinding applications (Db2 12 for z/OS).

Before you begin

To use bind avoidance, you must meet the following requirements:

- Your application must be written in VS COBOL, COBOL II, PL/I or ASM and run under a z/OS TSO environment. If you attempt to use bind avoidance on an unsupported language, such as C or FORTRAN, error message BND008E is returned.
- You must have access to the library that contains the DBRMs that were used during the previous bind operation.

  When you use bind avoidance, Db2 Admin Tool runs the Db2 precompiler for you and monitors its operation. To determine whether a bind is required, Db2 Admin Tool compares the results of the current precompile operation, the new DBRM, to the results of the previous precompile operation, the old DBRM.

If your installation does not have a policy to save DBRMs, you can request that Db2 Admin Tool generate a temporary working copy of the old DBRM from information in the Db2 catalog. This generated DBRM allows the comparison to proceed without interruption.

- If you plan to generate SMF records, the bind avoidance module must run from an APF-authorized library. Authorization is required to generate SMF records.

Before you begin, determine which of the following bind avoidance programs to run:

ADBBMAV
Use ADBBMAV after you have made changes to your application and need to first precompile and compile the application and then determine whether a bind is necessary.

ADBBMA3
Use ADBBMA3 if you have already successfully run ADBBMAV and therefore, precompiling the application again is not necessary. (ADBBMAV runs the Db2 precompiler.) In this case, you only need to determine whether a bind is required. For example, if you run ADBBMAV on an application and then migrate that application to a different subsystem, only a bind is necessary. In this case, you can run ADBBMA3 on the application.

If you plan to run ADBBMA3, you must bind this program with the plan name ADBBMA3 to connect to the Db2 subsystem. JCL for this bind operation is provided in member PLANBMA3 of SADBSAMP. Run this job after the Db2 Admin Tool Bind Manager packages are bound into a collection.

Restrictions:

- Bind avoidance does not support plans or packages that are prepared by using the LEVEL SQL processing option. The LEVEL option suppresses the generation of a unique consistency token. Because bind avoidance logic depends on these unique consistency tokens to make the decision about whether a bind is required, plans or packages with the LEVEL option are not supported.
- Bind avoidance does not support the use of the Db2 Coprocessor.

About this task

The ADBBMAV bind avoidance program does not connect to an active Db2 subsystem to validate DBRM and plan information. ADBBMAV assumes that the information in the DBRM library is identical to the information that is used by Db2. The only exception is if you choose to do the optional step of generating a temporary copy of an old DBRM. In this case, ADBBMAV connects to a subsystem.

Procedure

To determine whether applications need to be rebound, run one of the following programs:

- **ADBBMAV:** To run ADBBMAV, use JCL that is similar to the JCL that you use to call the precompiler with the following changes:
  - Change the EXEC statement to specify PGM=ADBBMAV instead of PGM=DSNHPC.
  - Specify where you want ADBBMAV to direct the report and messages. By default, the output is routed to SYSPRINT, which ADBBMAV shares with the precompiler. If you want the output to be routed to a data set other than the SYSPRINT data set, add the BNDLOG DD statement, as shown in the following example:

    ```
    //BNDLOG DD  SYSOUT=*
    ```

    This output data set can be a sequential data set or a member of a partitioned data set (PDS). If the output data set is a PDS, specify a member name in the JCL. If you specify a PDS with no member name, the data overwrites the PDS directory and corrupts the data set.
  - If you want ADBBMAV to generate SMF records to document the results of the bind decision, add a dummy DD statement with a DD name of SMF*xxx*, where *xxx* is the SMF record number (type).

xxx must be a value between 128 - 255. (Numbers outside this range are reserved by IBM.) For example:

```
//SMF128 DD DUMMY
```

You can use these SMF records to gauge the ratio of situations where a bind operation is required to situations where a bind operation is not required. You can also use these records for workload measurement and projections.

- If you want ADBBMAV to return a condition code other than 0 when the Db2 precompiler (program DSNHPC) issues one or more diagnostic warning messages, add a dummy DD statement with a DD name of SETRC*n*, where *n* specifies the return code. *n* can have a value of 2, 4, 6, or 8.

  For example, the following DD statement causes a condition code of 2 to be returned when DSNHPC issues one or more warning messages:

  ```
  //SETRC2 DD DUMMY
  ```

  By default, after ADBBMAV determines that a bind is not required, a return code of 0 is issued to control processing of subsequent job steps. A return code of 0 is issued even when the precompiler return code is 4, which indicates a warning message.
- If the old DBRM is not available and you want Db2 Admin Tool to generate a temporary working copy of the old DBRM, add a dummy DD statement with the DD name SSID*xxxx*, where *xxxx* is the Db2 subsystem name where DBRM information is to be extracted. *xxxx* can be 1 - 4 characters.

  For example, the following DD statement requests that a DBRM is generated based on the information in subsystem DSN1:

  ```
  //SSIDDSN1 DD DUMMY
  ```

  This temporary DBRM is deleted when the job step ends.
- If you want ADBBMAV to ignore DECLARE TABLE statements when comparing the old and new DBRMs, add a dummy DD statement with the DD name SKIPDECL. For example:

  ```
  //SKIPDECL DD DUMMY
  ```
- If you want ADBBMAV to not compare the coded character set identifiers (CCSIDs) of the old DBRMs with the CCSIDs of the new DBRMs, add a dummy DD statement with the DD name NOCC. For example:

  ```
  //NOCC DD DUMMY
  ```

  By default, ADBBMAV compares these CCSIDs. If they do not match, ADBBMAV considers the DBRMs to be a mismatch and requires a bind.

When you submit the JCL to run ADBBMAV, Db2 Admin Tool runs the precompiler for you, determines whether a bind operation is required, and then provides one of the following return codes:

Return code	Meaning
0 (zero)	You can skip the bind operation.
4	A bind operation is required.

If you requested SMF records, they are generated in the following format:

Table 46. SMF record format

Offset (decimal)	Field Name	Length (bytes)	Notes
18	JOBNAME	8	
26	DBRMNAME	8	

Table 46. SMF record format (continued)

Offset (decimal)	Field Name	Length (bytes)	Notes
34	USERID	7	
41	AVBFLAG	1	**Y** Successful. No bind required. **N** Unsuccessful. A bind is required.
42	TIMESTAMP	26	Old timestamp value. This value is meaningful only if `AVBFLAG='Y'`.

- **ADBBMA3:** To run ADBBMA3, use member ADBBMA3 in the SADBSAMP library as a model for creating your own JCL.

  Requirement: To run ADBBMA3, you must have already run ADBBMAV for the application. See "ADBBMA3" on page 889.

  Include the following DD statements as needed:

DD statement	Identifies
STEPLIB	The load library for Db2 Admin Tool and the Db2 SDSNLOAD library.
BINDOUT	The output data set to which ADBBMA3 is to write any required BIND commands.
BINDREJ	**Optional** The output data set to which ADBBMA3 is to write the BIND commands that are not required.
SYSPRINT	**Optional** The output data set to which ADBBMA3 is to write any error messages and program identification.
DBRMIN	The DBRMLIB data set that contains the DBRMs to be processed.
BINDIN	The input data set that contains the BIND commands for the application with the DSN command. The DSN command identifies which Db2 subsystem to check for existing entries in SYSIBM.SYSPACKAGE.

  ADBBMA3 identifies the program (DBRM) from the BIND commands in BINDIN. ADBBMA3 then checks the Db2 catalog table SYSIBM.SYSPACKAGE to determine if the program was already bound to that collection with the same program name (DBRM) and consistency token (CONTOKEN).

  ADBBMA3 sets the return code as follows to indicate whether a bind operation is necessary:

Return code	Meaning
0	At least one of the BIND commands (in BINDOUT) must be processed. If the BIND commands in the BINDIN data set include a BIND PLAN with PKLIST, that command is always processed. If the DBRM does not exist in the library that is specified in the DBRMIN DD statement, the BIND commands are passed to BINDOUT, and the return code is 0.
4	None of the BIND commands need to be processed. The only contents of the BINDOUT data set are the DSN command and END.
8	At least one error was detected. Review SYSPRINT. If SYSPRINT is not allocated, rerun the job with a SYSPRINT DD statement and then review SYSPRINT.

You can optionally pass the following parameter when calling ADBBMA3:

UNIQUE-VERSION
Specifies that a BIND command is not to be generated when the package consistency token (CONTOKEN) is changed but the Db2 catalog and DBRM have the same package version. In this case, ADBBMA3 ends with return code 8.

Example: The following example JCL EXEC statement shows how to pass the UNIQUE-VERSION option when calling ADBBMA3:

```
//BNDA3 EXEC PGM=ADBBMA3,PARM=('OPTION UNIQUE-VERSION ')
```

Related reference
Descriptions of SQL processing options (Db2 12 for z/OS)

Activating IBM Db2 AI for z/OS

IBM Db2 AI for z/OS (Db2ZAI) leverages machine learning technology to empower the Db2 for z/OS optimizer to determine the best-performing query access paths, based on your workload characteristics.

Procedure

To activate Db2ZAI:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option 2M, and press Enter.
3. On the **Display/Manage Machine Learning (ADB2Z2M)** panel, specify option 2, and press Enter.

```
ADB2Z2M n               DC1A  Display/Manage Machine Learning              14:30
Option ===>

   1 - Display ML
   2 - Start ML
   3 - Stop ML

Note: the Db2 commands will be issued immediately
```

The Db2 command STA ML is issued immediately unless you have the preference set to be prompted before executing Db2 commands. In this case, the **Statement Execution Prompt (ADB2PSTM)** panel is displayed and you can specify 1 to execute the command.

The STA ML command activates Db2ZAI on the subsystem or, in a data sharing environment, all members in the data sharing group.

4. Optional: To confirm that Db2ZAI is activated, exit back to the **Display/Manage Machine Learning (ADB2Z2M)** panel, specify option 1, and press Enter.

   The Db2 command DISPLAY ML is issued, and the output from the command is displayed to indicate that Db2ZAI is active:

```
 ADB2DB2O  ------- DC1A Browse DB2 Command Output ---  Line 00000000 Col 001 080

 -DIS ML


********************************* Top of Data *********************************
DSNX611I  !DC1A DSNXODPM DISPLAY ML REPORT FOLLOWS:
           STATUS     = STARTED
             - ML Daemon > RUNNING
             - Heartbeat monitoring > RUNNING*
             - Execution History Pushout > RUNNING
             - AccessPath History Pushout > RUNNING
             - HV Model Training > RUNNING
             - OFNR Model Training > RUNNING
             - Model Statistics Pushout > RUNNING
             - Performance Data Processing > RUNNING
             - Cleanup Processing > RUNNING
             - Regression Retraining > RUNNING*
             - Training Status Processing > RUNNING
             - ML Statistics Processing > RUNNING
             - ML Scheduler > RUNNING
```

What to do next

If you later want to stop Db2ZAI, return to the **Display/Manage Machine Learning (ADB2Z2M)** panel and specify option 3 to issue the Db2 command STOP ML.

Related tasks

“Changing Db2 Admin Tool prompt options” on page 243

Related information

DISPLAY ML command (Db2 12 for z/OS)

START ML command (Db2 12 for z/OS)

STOP ML command (Db2 12 for z/OS)

Authorization switching

Authorization switching is a facility within Db2 Admin Tool that is used to execute DDL and DCL under the authority of another user. The facility does not cover other statement types, including Db2 utility statements and DSN subcommands such as FREE PACKAGE and BIND PLAN.

Deprecation notice: Authorization switching is deprecated in Db2 Admin Tool. For more information, see “Deprecated functions and functions that are no longer supported in Db2 Admin Tool 12.1” on page 64.

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

Alter Tablespace ALT, Alter Table ALT, WSLs, Change Management, Change Management batch, and IBM Db2 Object Comparison Tool for z/OS make use of authorization switching. These commands enable you to redefine a table space or a table. Because this action also requires the object to be dropped, objects that are dependent on the target object are also dropped. Authorizations to those objects and dependent objects are lost.

Db2 Admin Tool generates the DDL and DCL necessary to rebuild the altered objects and to restore the dependent objects and authorities. However, the job submitter might have authority to re-create the target objects, but not the authority to re-create dependent objects or to grant authorities to the dependent objects. In this case, you can enable a job submitter to use an ID that has the necessary authority to execute the DDL to rebuild the objects.

The statements that you can run with the auth-switch ID depend on your authority as defined in the RACF profile that protects the resource. If you have READ authority, the authorization switching function follows these rules and protections:

- Only certain DDL statements can be run using the auth-switch ID. ALTER (TABLE, MASK, PERMISSION, FUNCTION, and TRIGGER), COMMENT, LABEL, CREATE, SET, GRANT (except system privilege) are auth-switch eligible statements. DROP statements, for example, are always run using the submitter's authority.
- Any DDL that has been manually added to the file or that has been edited can be run only under the submitter's authority.
- COMMIT statements can be added where appropriate.
- The DDL must be run within 8 days of being created.
- If ineligible statements are encountered, Db2 Admin Tool will switch out of the requesting auth-switch ID and back into the auth-switch ID when an eligible statement is encountered.

If the job submitter has ALTER authority to the RACF profile that protects the resource, all DDL and DCL statements are run using the auth-switch ID. The rules and protection mechanisms for READ authority do not apply for ALTER authority.

When authorization switching is enabled, the batch job panels for Alter Tablespace AL and Alter Table ALT have an additional input field called **Authorization Switch ID** . Use this field to enter the auth-switch ID to be used to run the eligible statements in the file that contains the DDL and DCL statements.

The DDL that is generated by the batch jobs for these two functions is prepared for authorization switching; that is, it contains functional comments that other Db2 Admin Tool components use with authorization switching.

The value <SQLID> is similar to the value SQLID except that you do not need to specify a specific auth-switch ID. If the value <SQLID> is specified in the Authorization Switch ID field, SQLID auth-switching DDL is generated. The SQLID auth-switching DDL automatically implements authorization switching. When the WSL runs, Db2 Admin Tool can use dynamic auth-switching to drop and re-create objects by using the existing owner of the objects. During the running of a procedure, the WSL detects the need for authorization switching and provides the required auth-switch ID.

If the special value <NONE> is specified in the **Authorization Switch ID** field, the DDL is not prepared to be used with authorization switching, but an authorized ID can run the DDL. For example, the authorized ID can run the DDL using ADBTEP2.

If an authorization switch ID is not specified, and you specify Y in **As work statement list**, the work statement list does not produce DDL that is capable of authorization switching.

Tip: Carefully preserve the original DDL file until the objects and dependencies are restored. After the object is dropped, the ADB2GEN process cannot be used to regenerate the original environment. Running the ADB2GEN step again without proper care can overwrite the original DDL file, making reconstruction difficult.

The batch program, ADBTEPA, runs the DDL, either under the authority of the submitter or under the auth-switch ID authority. Two input parameters are required for authorization switching. These parameters are specified one-per line on the ADBOPT DD card in the ADBTEPA step.

Example

```
//CREATE   EXEC PGM=ADBTEPA,DYNAMNBR=100,
//  PARM='/SSID(DSN8),WORKLIST(TESTYA)'
//STEPLIB  DD DISP=SHR,DSN=DMTOOL.SADBLINK
//         DD DISP=SHR,DSN=DSN810.SDSNEXIT
//         DD DISP=SHR,DSN=DSN810.SDSNLOAD
//SYSTSPRT DD SYSOUT=*
//ADBPRINT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
```

```
//ADBOPT   DD *
 PLAN=ADBTEPA
 AUTH_SWITCH_USERID=SYSADMZ1
/*
```

The PLAN parameter is required by ADBTEPA, even when an auth-switch ID is not provided on the batch job panels. The AUTH_SWITCH_USERID parameter is generated, either as functional input when an ID is provided on the panel, or it is a comment without a value. If Db2 Admin Tool Authorization Switching is determined to be necessary after the JCL is built, you can make the parameter active (remove the comment) and specify a suitable auth-switch ID.

To use Db2 Admin Tool Authorization Switching, the job submitter must have access to the following two separate entities:

- The plan that is passed to program ADBTEPA using the ADBOPT parameter PLAN
- A RACF profile that protects a special resource

If the submitter has READ authority to the RACF profile, only certain DDL statements are executed using the authorization switch ID authority. DROP statements, for example, are always executed using the submitter's authority. If the job submitter has ALTER authority to the RACF profile that protects the resource, all DDL and DCL statements are run using the auth-switch ID. The rules and protection mechanisms that apply to READ authority do not apply to ALTER authority.

Only certain DDL statements are executed using the authorization switch ID authority. DROP statements, for example, are always executed using the submitter's authority. Any DDL that has been manually added to the file or that has been edited can be executed only under the submitter's authority. COMMIT statements can be added where appropriate.

Because the DDL contains SET CURRENT SQLID statements, the ID that runs the statement must have the intended SQLID as one of its secondary authorization IDs, or have SYSADM authority. If you want to suppress the generation of SET CURRENT SQLID statements, specify the value for Run SQLID as <NONE>. You can specify a RUN SQLID value in Db2 Admin Tool functions such as GEN, ALT, Migrate, and Change Management.

Tips:

- Provide the authorization switch ID with SYSADM authority to successfully execute all statements within the DDL file, including the SET CURRENT SQLID statements.
- When you specify <SQLID> as the auth-switch ID, the RUN SQLID field must be blank.
- If you use the authorization switch with Db2 sample sign on exit, you must complete one of the following steps:
  - Run the job by using an authorization switch ID that has SYSADM authority or that is connected to a group that has SYSADM authority.
  - Define the owner of the objects as a RACF group and then run the job by using an authorization switch ID that is connected to the group ID.

  For either of the preceding steps, define the authorization switch ID as a RACF user ID without a known password and with a password that never expires.

Recommendation: If you want objects to be owned or managed by a secondary authorization ID (or RACF group) with SYSADM authority, use Run SQLID. When you specify a value for Run SQLID, Db2 Admin Tool adds a SET CURRENT SQLID=*runsqlid* statement to the beginning of the change and runs everything with that SQLID. This scenario requires that the Db2 security exit define the SQLID as a secondary authorization ID for the user that submits the job. For a RACF group, the user must be connected to that group.

Alternatively, use authorization switching when you want to give a user with the required RACF authorizations the privilege to connect to Db2 with a different user ID and without a password so that they can recreate objects with DDL that was generated by Db2 Admin Tool. When you use authorization switching, the DDL statements are protected by a key so that they cannot be changed. This scenario allows database administrators (DBAs) without the SYSADM privilege to recreate objects. For example, they can recreate views after a table is dropped and recreated. In this case, the database administrator uses the privileges of the original object creator by connecting to Db2 with the required user ID.

Finally, you can use a trusted context in Db2 Admin Tool if you want changes to be made only by using a trusted context. In this case, the DBAs do not have SYSADM privilege by using a privilege on a secondary authorization ID.

SQLID authorization switching

SQLID authorization switching enables you to use authorization switching for tasks that require two or more authorization IDs. SQLID authorization switching can be used with WSLs, Change Management, Change Management batch, IBM Db2 Object Comparison Tool for z/OS, and ALT.

To use SQLID auth-switching, you must specify an `<SQLID>` instead of a regular authorization ID. When you specify an SQLID as the auth-switch ID, the RUN SQLID field must be blank.

The SQLID authorization switching function follows these rules and protections:

- The SQLID must be verified for SET CURRENT SQLID to be executed.
- To verify the SQLID, execute the following statement using the submitter's ID:

  ```
  SELECT VERIFY_GROUP_FOR_USER(USER,:newsqlid)
   FROM SYSIBM.SYSDUMMY1
  ```

  The new SQLID is the ID used in SET CURRENT SQLID. When the SQLID you specify is verified, the statement returns 1, and SET CURRENT SQLID is executed
- If the ID is not verified, all statements will be executed using the submitter's ID, with these exceptions: CREATE and SET CURRENT SQLID statements.
- The CREATE statement will use the ID that was used in the last SET CURRENT SQLID statement. The ID can be a verified ID or a non-verified ID. If no SET CURRENT SQLID is provided, the submitter's ID is used.
- If a statement fails to execute, an authorization error such as -551 is issued, and the operation is performed using the object owner's authorization ID. There are exceptions where the object owner's authorization ID is not used as the retry ID. One exception is when the retry ID is used for GRANT and REVOKE on a view that uses the last ID specified in a PATHSCHEMAS. The other exception is when a retry ID is used for a CREATE statement.
- A CREATE statement retry is performed for Alias, Table, and View objects only when the object is qualified. The retry will use the submitter's ID as the authorization ID. Retry ID for a view uses the last ID specified in the CURRENT PATH or qualifier.
- If no SET CURRENT SQLID is provided in a WSL, ADBTEP2 will use the submitter's ID for alias, view, and table objects only when the object name is qualified; the retry for this case will use a qualifier as an authorization ID.

Using WSL authorization switching

WSL *authorization switching* enables you to run a regular WSL by using an auth-switch ID. You can only execute WSL auth-switching from the WSL pane.

Procedure

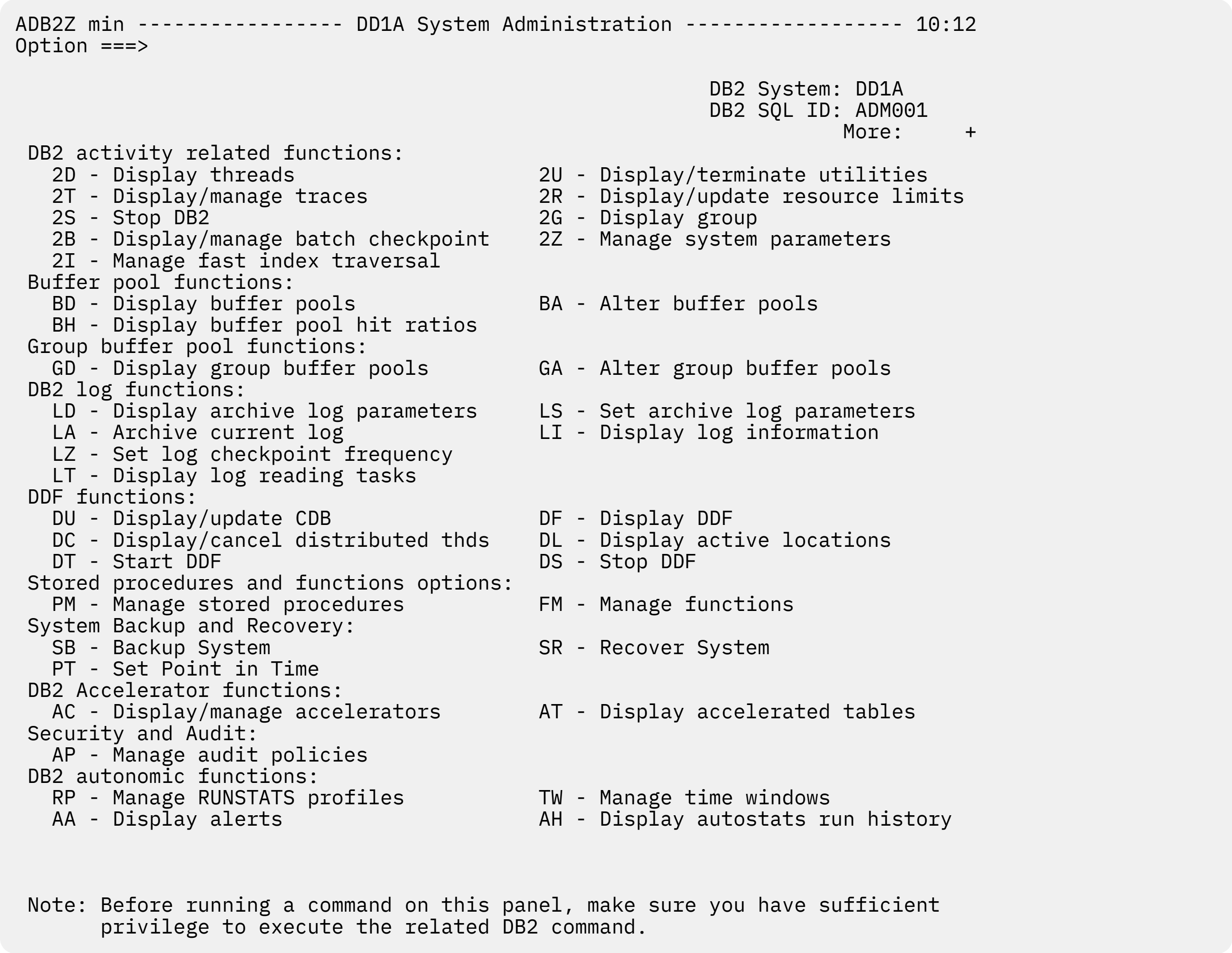
1. On the **Change Options Common to Change Functions** panel (ADB2PCO), set Enable WSL authorization switching to YES.
2. On the **Work Statement List Library** panel (ADB2W1), enter the R (RUN) line command to run a WSL. The **Authorization Parameters** panel (ADBPWLA) will appear.
3. On the **Authorization Parameters** panel, enter an *auth-switch ID* that you want to use as the primary authorization ID. When you specify the *auth-switch ID*, JCL that is eligible for WSL authorization switching is generated.

Note: You must have ALTER authority to the RACF profile to use the WSL authorization switching function. You cannot use SQLID auth-switching along with WSL auth-switching.

Chapter 5. Db2 systems administration

You can use Db2 Admin Tool to administer your Db2 systems. For example, you can display threads and terminate utilities.

The system administration tasks that are supported by Db2 Admin Tool are listed on the **System Administration (ADB2Z)** panel, as shown in the following figure. To display this panel, specify option Z on the **DB2 Administration Menu (ADB2)** panel.



```
ADB2Z min ----------------- DD1A System Administration ------------------ 10:12
Option ===>
                                                         DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
                                                                    More:     +
 DB2 activity related functions:
   2D - Display threads                     2U - Display/terminate utilities
   2T - Display/manage traces               2R - Display/update resource limits
   2S - Stop DB2                            2G - Display group
   2B - Display/manage batch checkpoint     2Z - Manage system parameters
   2I - Manage fast index traversal
 Buffer pool functions:
   BD - Display buffer pools                BA - Alter buffer pools
   BH - Display buffer pool hit ratios
 Group buffer pool functions:
   GD - Display group buffer pools          GA - Alter group buffer pools
 DB2 log functions:
   LD - Display archive log parameters      LS - Set archive log parameters
   LA - Archive current log                 LI - Display log information
   LZ - Set log checkpoint frequency
   LT - Display log reading tasks
 DDF functions:
   DU - Display/update CDB                  DF - Display DDF
   DC - Display/cancel distributed thds     DL - Display active locations
   DT - Start DDF                           DS - Stop DDF
 Stored procedures and functions options:
   PM - Manage stored procedures            FM - Manage functions
 System Backup and Recovery:
   SB - Backup System                       SR - Recover System
   PT - Set Point in Time
 DB2 Accelerator functions:
   AC - Display/manage accelerators         AT - Display accelerated tables
 Security and Audit:
   AP - Manage audit policies
 DB2 autonomic functions:
   RP - Manage RUNSTATS profiles            TW - Manage time windows
   AA - Display alerts                      AH - Display autostats run history


 Note: Before running a command on this panel, make sure you have sufficient
       privilege to execute the related DB2 command.
```

*Figure 421. **System Administration (ADB2Z)** panel*

You can choose any of the following options:

2D – Display threads
Displays the current status of Db2 threads.

2U – Display/terminate utilities
Displays the status of utility jobs or terminates utility jobs.

2T – Display/manage traces
Displays, starts, or stops traces.

2S – Stop DB2
Stops the Db2 subsystem.

2G – Display Group
Displays information about the data sharing group to which the Db2 subsystem belongs.

2B – Display/manage batch checkpoint table
Displays and manages the checkpoint table (ADBCHKPT) that is associated with batch jobs that are running ADBTEP2. You can use ADBTEP2 to restart or resume execution of an input stream of SQL statements at an intermediate point in case one of the statements fails.

2R – Display/update resource limits (RLIMIT)
Displays or stops the resource limit (RLIMIT) facility or updates the RLIMIT tables that are created in the system.

2M - Display/manage machine learning
Displays, starts, and stops Db2 machine learning.

2I - Manage fast index traversal
Displays memory usage for fast index traversal and controls which indexes use fast index traversal.

2Z – Manage system parameters
Dynamically manages subsystem parameters.

BD – Display buffer pools
Displays the current status of one or more active or inactive buffer pools.

BA – Alter buffer pools
Alters the attributes of active or inactive buffer pools.

BH – Display buffer pool hit ratios
Displays the hit ratios for the buffer pools.

GD – Display group buffer pools
Displays the group buffer pools for Db2 data sharing. This field is for a data sharing environment only.

GA – Alter group buffer pools
Alters the group buffer pools for Db2 data sharing. This field is for a data sharing environment only.

LI – Display log information
Displays information about the Db2 log.

LD – Display archive log parameters
Displays information about the input archive log.

LA – Archive current log
Archives the current Db2 log.

LS – Set archive log parameters
Sets the upper limit for the number of and the deallocation time of tape units for the archive log.

LZ – Set log checkpoint frequency
Sets the Db2 system checkpoint frequency.

LT - Display log reading tasks
Displays statistics about any log reading tasks that are currently running.

DU – Display/update CDB
Displays or updates a table in the communications database (CDB).

DF – Display DDF
Displays the status and configuration of the distributed data facility (DDF).

DC – Display/cancel distributed thds
Displays or cancels processing for threads that originate locally and access remote data, or originate remotely and access local data.

DL – Display active locations
Display statistics about threads with a distributed relationship or displays conversation information about Db2 system threads that interact with VTAM®.

DT – Start DDF
Starts the DDF if it has not already been started.

DS – Stop DDF
Stops the DDF if it has already been started.

PM – Manage stored procedures
Manages Db2 stored procedures.

FM – Manage functions
Manages Db2 user-defined functions.

SB – Backup System
Backs up the Db2 subsystem.

SR – Recover System
Sets up a job to recover the Db2 subsystem to a point in time.

PT – Set point in time
Sets a particular time at which to recover the Db2 subsystem.

AC - Display/manage accelerators
Displays or updates Db2 accelerators.

AT - Display accelerated tables
Displays the Db2 tables that are considered for query offloading to the accelerators.

AP - Manage audit policies
Displays and manages security audit policies for tables or aliases.

RP - Manage RUNSTATS profiles
Displays and manages RUNSTATS profiles for table objects.

TW - Manage time windows
Displays and manages time windows, when the autonomic collection of statistics is allowed.

AA - Display alerts
Displays alerts that are generated during the autonomic collection of statistics.

AH - Display autostats run history
Displays RUNSTATS history that is generated during autonomic collection of statistics.

Displaying threads

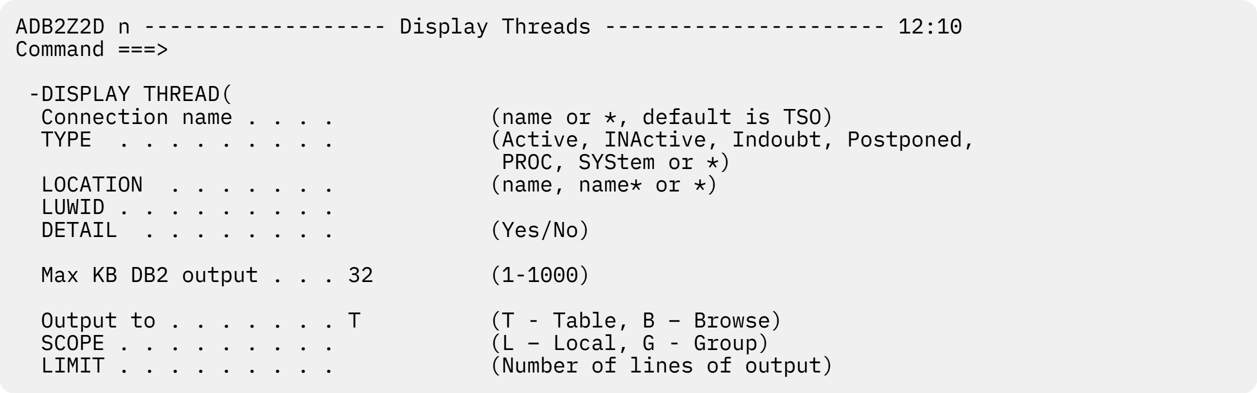
You can display the current status of Db2 threads.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option 2D, and press Enter.

   The **Display Threads (ADB2Z2D)** panel is displayed, as shown in the following figure.



```
ADB2Z2D n ------------------- Display Threads ---------------------- 12:10
Command ===>

 -DISPLAY THREAD(
  Connection name . . . .            (name or *, default is TSO)
  TYPE  . . . . . . . . .            (Active, INActive, Indoubt, Postponed,
                                      PROC, SYStem or *)
  LOCATION  . . . . . . .            (name, name* or *)
  LUWID . . . . . . . . .
  DETAIL  . . . . . . . .            (Yes/No)

  Max KB DB2 output . . . 32         (1-1000)

  Output to . . . . . . . T          (T - Table, B – Browse)
  SCOPE . . . . . . . . .            (L – Local, G - Group)
  LIMIT . . . . . . . . .            (Number of lines of output)
```

*Figure 422. **Display Threads (ADB2Z2D)** panel*

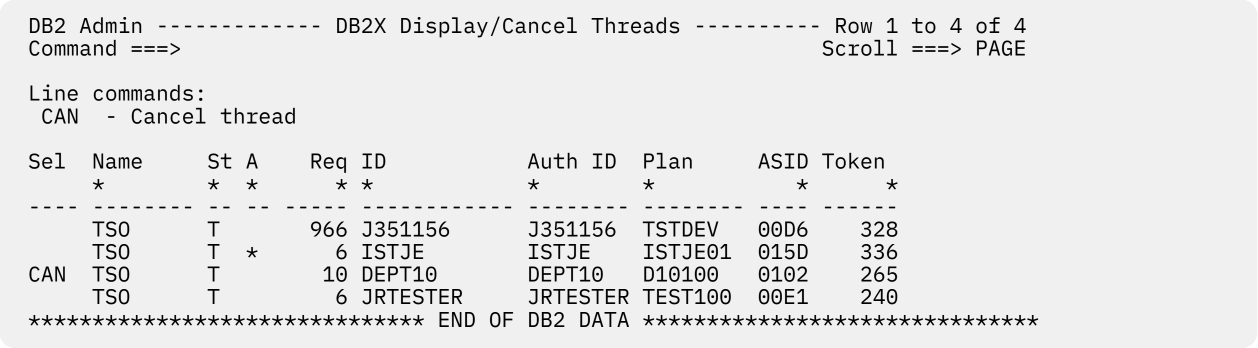
3. Specify values for the fields on the panel that you need, and press Enter.

   Db2 Admin Tool issues the Db2 DISPLAY THREAD command.

The information that Db2 Admin Tool returns to you from the command is in ISPF browse format or in a table display panel, depending on what you specify in the **Output to** field.

If you choose to display the Db2 threads on a table display panel, the **Display/Cancel Threads (ADB2Z2D2)** panel is displayed, as shown in the following figure. On this panel, you can cancel Db2 threads.

Restriction: You cannot cancel a thread that is running under the active user ID. An asterisk in the A column indicates which thread is associated with the active user ID.



```
DB2 Admin ------------- DB2X Display/Cancel Threads ---------- Row 1 to 4 of 4
Command ===>                                                  Scroll ===> PAGE

Line commands:
 CAN  - Cancel thread

Sel  Name     St A    Req ID           Auth ID  Plan     ASID Token
     *        *  *      * *            *        *          *      *
---- -------- -- -- ----- ------------ -------- -------- ---- ------
     TSO      T        966 J351156      J351156  TSTDEV   00D6    328
     TSO      T  *       6 ISTJE        ISTJE    ISTJE01  015D    336
CAN  TSO      T         10 DEPT10       DEPT10   D10100   0102    265
     TSO      T          6 JRTESTER     JRTESTER TEST100  00E1    240
******************************* END OF DB2 DATA *******************************
```

*Figure 423. **Display/Cancel Threads (ADB2Z2D2)** panel*

Displaying or terminating utilities

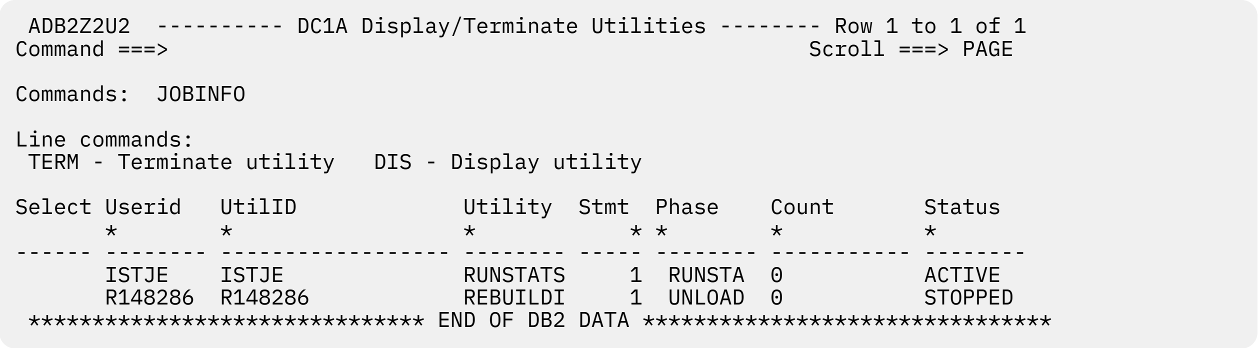
You can display the status of a Db2 utility job to determine if it is active, stopped, or terminated and the phase in which the utility is currently executing or stopped. If you choose to terminate a utility, use caution. You cannot restart a terminated utility job, and objects might be left in an indeterminate state.

Procedure

To display or terminate utilities:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option 2U, and press Enter.

   The **Display or Terminate Utilities (ADB2Z2U2)** panel is displayed:



```
 ADB2Z2U2  ---------- DC1A Display/Terminate Utilities -------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Commands:  JOBINFO

Line commands:
 TERM - Terminate utility   DIS - Display utility

Select Userid   UtilID             Utility  Stmt  Phase    Count       Status
       *        *                  *            * *        *           *
------ -------- ------------------ -------- ----- -------- ----------- --------
       ISTJE    ISTJE              RUNSTATS     1  RUNSTA  0           ACTIVE
       R148286  R148286            REBUILDI     1  UNLOAD  0           STOPPED
 ******************************* END OF DB2 DATA ********************************
```

*Figure 424. **Display or Terminate Utilities (ADB2Z2U2)** panel*

The following fields are available on this panel:

SELECT
: Input field where you can enter one of the line commands listed on the panel

UTILITY
: Name of the utility

USERID
: User ID of the person who is running the utility

UTIL ID
Utility identifier

STMT
Number of the utility statements being processed

PHASE
Current phase of the utility, such as RELOAD

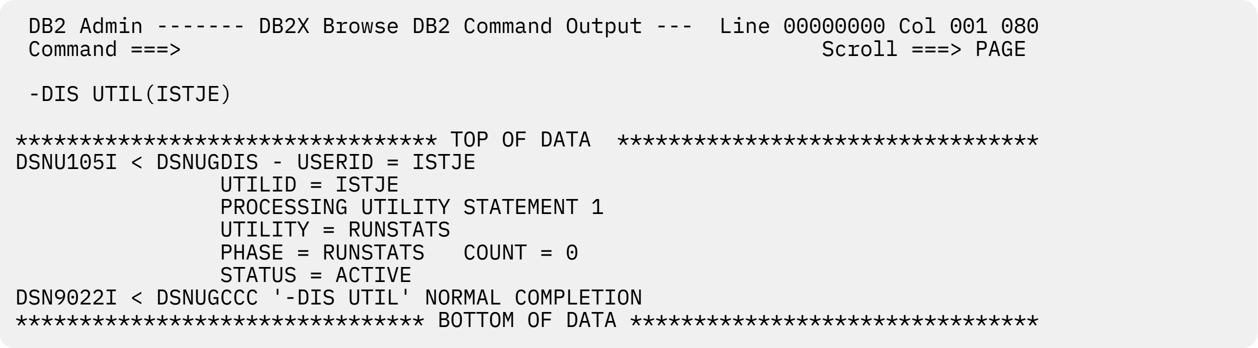
COUNT
Depending on the utility that is currently running, the number of rows, pages, or page sets being processed

STATUS
Status of the utility, such as ACTIVE

3. Take one of the following actions:

   - To display the status of a utility, issue the DIS line command next to the utility for which you want to display the status, and press Enter. Db2 Admin Tool issues the Db2 command DISPLAY UTILITY.
   - To terminate the utility, issue the TERM line command next to the utility that you want to terminate, and press Enter. Db2 Admin Tool issues the Db2 command TERMINATE UTILITY.

   The information that Db2 Admin Tool returns to you from the commands is in ISPF browse format. The following figure shows the type of information Db2 Admin Tool returns when you issue the DIS line command:



```
 DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

 -DIS UTIL(ISTJE)

********************************* TOP OF DATA  *********************************
DSNU105I < DSNUGDIS - USERID = ISTJE
                UTILID = ISTJE
                PROCESSING UTILITY STATEMENT 1
                UTILITY = RUNSTATS
                PHASE = RUNSTATS   COUNT = 0
                STATUS = ACTIVE
DSN9022I < DSNUGCCC '-DIS UTIL' NORMAL COMPLETION
******************************** BOTTOM OF DATA ********************************
```

Figure 425. ***Display Utilities (ADB2DB2O)*** *panel*

Managing RUNSTATS profiles

A *RUNSTATS profile*, also called a *statistics profile*, is a saved set of options for the RUNSTATS utility or inline statistics from the LOAD or REORG TABLESPACE utilities.

Procedure

To manage RUNSTATS profiles:

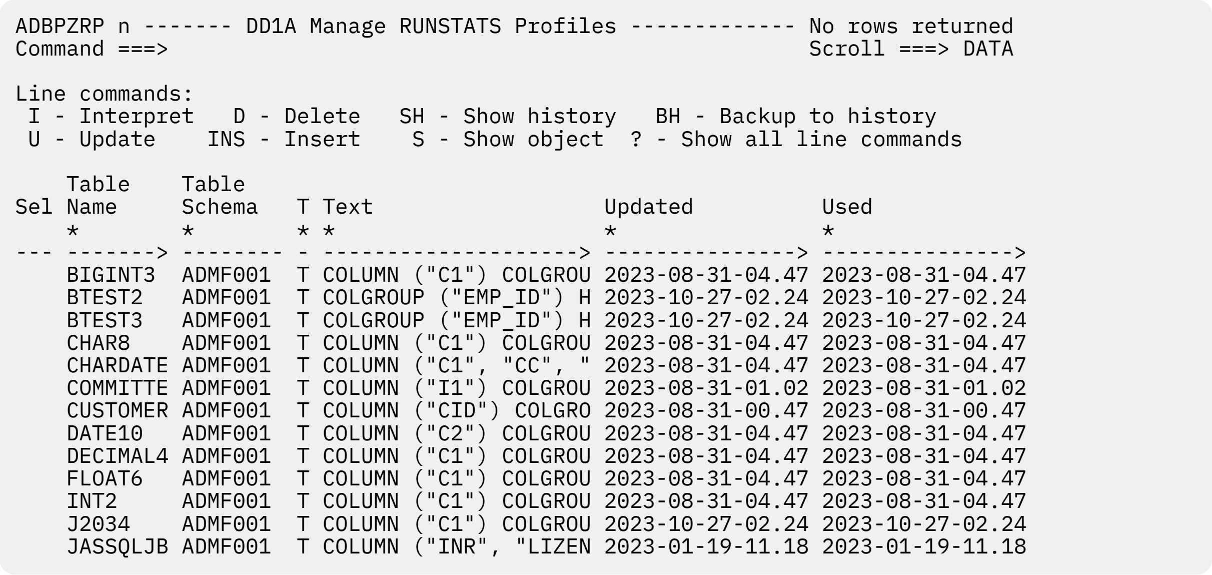
1. Navigate to the **Manage RUNSTATS Profiles (ADBPZRP)** panel by completing one of the following procedures:

   To view a list of profiles for up to 1,000 tables:

   a. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
   b. On the **System Administration (ADB2Z)** panel, specify option RP, and press Enter.

   The **Manage RUNSTATS Profiles (ADBPZRP)** panel lists the existing RUNSTATS profiles:

   Note: If profiles exist for more than 1,000 tables, the table you want might not be listed. In this case, use the second procedure ("To view the profile for a specific table").



```
ADBPZRP n ------- DD1A Manage RUNSTATS Profiles ------------- No rows returned
Command ===>                                                  Scroll ===> DATA

Line commands:
 I - Interpret   D - Delete   SH - Show history   BH - Backup to history
 U - Update    INS - Insert    S - Show object  ? - Show all line commands

    Table    Table
Sel Name     Schema   T Text                  Updated          Used
    *        *        * *                     *                *
--- -------> -------- - --------------------> ---------------> --------------->
    BIGINT3  ADMF001  T COLUMN ("C1") COLGROU 2023-08-31-04.47 2023-08-31-04.47
    BTEST2   ADMF001  T COLGROUP ("EMP_ID") H 2023-10-27-02.24 2023-10-27-02.24
    BTEST3   ADMF001  T COLGROUP ("EMP_ID") H 2023-10-27-02.24 2023-10-27-02.24
    CHAR8    ADMF001  T COLUMN ("C1") COLGROU 2023-08-31-04.47 2023-08-31-04.47
    CHARDATE ADMF001  T COLUMN ("C1", "CC", " 2023-08-31-04.47 2023-08-31-04.47
    COMMITTE ADMF001  T COLUMN ("I1") COLGROU 2023-08-31-01.02 2023-08-31-01.02
    CUSTOMER ADMF001  T COLUMN ("CID") COLGRO 2023-08-31-00.47 2023-08-31-00.47
    DATE10   ADMF001  T COLUMN ("C2") COLGROU 2023-08-31-04.47 2023-08-31-04.47
    DECIMAL4 ADMF001  T COLUMN ("C1") COLGROU 2023-08-31-04.47 2023-08-31-04.47
    FLOAT6   ADMF001  T COLUMN ("C1") COLGROU 2023-08-31-04.47 2023-08-31-04.47
    INT2     ADMF001  T COLUMN ("C1") COLGROU 2023-08-31-04.47 2023-08-31-04.47
    J2034    ADMF001  T COLUMN ("C1") COLGROU 2023-10-27-02.24 2023-10-27-02.24
    JASSQLJB ADMF001  T COLUMN ("INR", "LIZEN 2023-01-19-11.18 2023-01-19-11.18
```

Figure 426. ***Manage RUNSTATS Profiles (ADBPZRP)*** *panel*

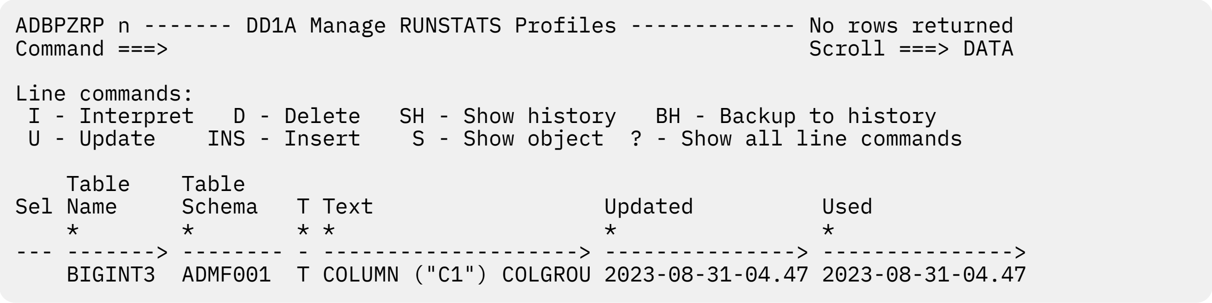
To view the profile for a specific table:

a. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.

b. On the **System Catalog (ADB21)** panel, specify option T and any selection criteria, and press Enter.

c. On the **Tables, Views, and Aliases (ADB21T)** panel, issue the RP line command next to the desired table.

The **Manage RUNSTATS Profiles (ADBPZRP)** panel lists the RUNSTATS profile for that table:



```
ADBPZRP n ------- DD1A Manage RUNSTATS Profiles ------------- No rows returned
Command ===>                                                  Scroll ===> DATA

Line commands:
 I - Interpret   D - Delete   SH - Show history   BH - Backup to history
 U - Update    INS - Insert    S - Show object  ? - Show all line commands

    Table    Table
Sel Name     Schema   T Text                  Updated          Used
    *        *        * *                     *                *
--- -------> -------- - --------------------> ---------------> --------------->
    BIGINT3  ADMF001  T COLUMN ("C1") COLGROU 2023-08-31-04.47 2023-08-31-04.47
```

Figure 427. ***Manage RUNSTATS Profiles (ADBPZRP)*** *panel*

2. Use the line commands to take any necessary actions.

   For example:

   To add a new profile:
   Specify the INS line command in any row.

   To edit the profile text:
   Specify the U line command next to the profile that you want to edit. On the subsequent panel, you can edit the profile text.

Related reference

Statistics profiles (Db2 12 for z/OS)

Managing traces

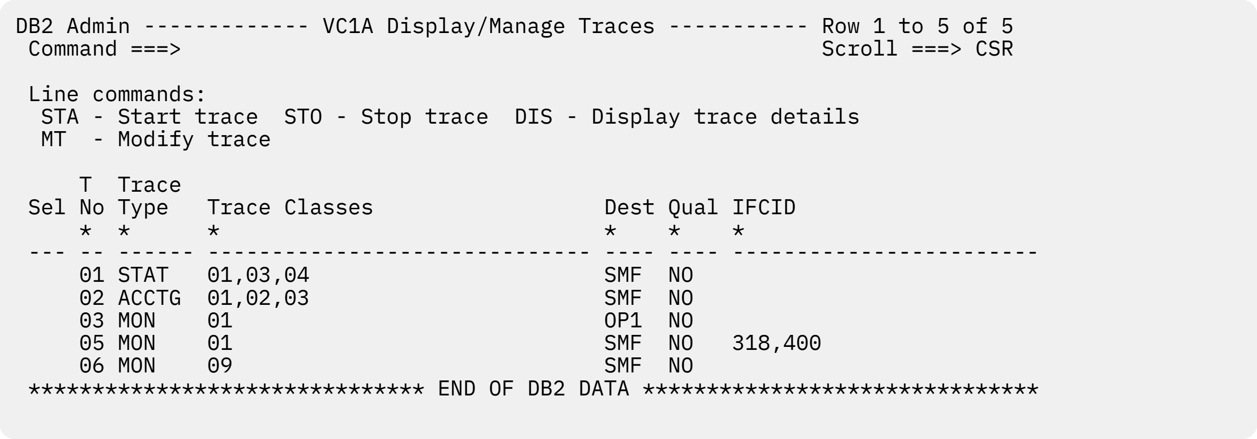
Traces record Db2 data and events. You can use Db2 Admin Tool to display, start, or stop traces.

Procedure

To manage traces:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option 2T, and press Enter.

   The **Display/Manage Traces (ADB2Z2T2)** panel is displayed:



```
DB2 Admin ------------- VC1A Display/Manage Traces ----------- Row 1 to 5 of 5
 Command ===>                                                  Scroll ===> CSR

 Line commands:
  STA - Start trace  STO - Stop trace  DIS - Display trace details
  MT  - Modify trace

     T  Trace
 Sel No Type   Trace Classes                  Dest Qual IFCID
     *  *      *                              *    *    *
 --- -- ------ ------------------------------ ---- ---- ------------------------
     01 STAT   01,03,04                       SMF  NO
     02 ACCTG  01,02,03                       SMF  NO
     03 MON    01                             OP1  NO
     05 MON    01                             SMF  NO   318,400
     06 MON    09                             SMF  NO
 ******************************* END OF DB2 DATA *******************************
```

   Note: You can also open the **Display/Manage Traces (ADB2Z2T2)** panel from the **Display/Manage Statement Cache Table (ADB2ES)** panel by selecting option 1.

   This panel contains the following fields:

   Sel
   Selection field. Use this field to enter one of the line commands that is listed on the panel.

   T No
   Trace number.

   Trace Type
   Trace type.

   Trace Classes
   Trace classes that are active for the trace.

   Dest
   Destination for the trace.

   Qual
   An indication of whether the trace is further qualified.

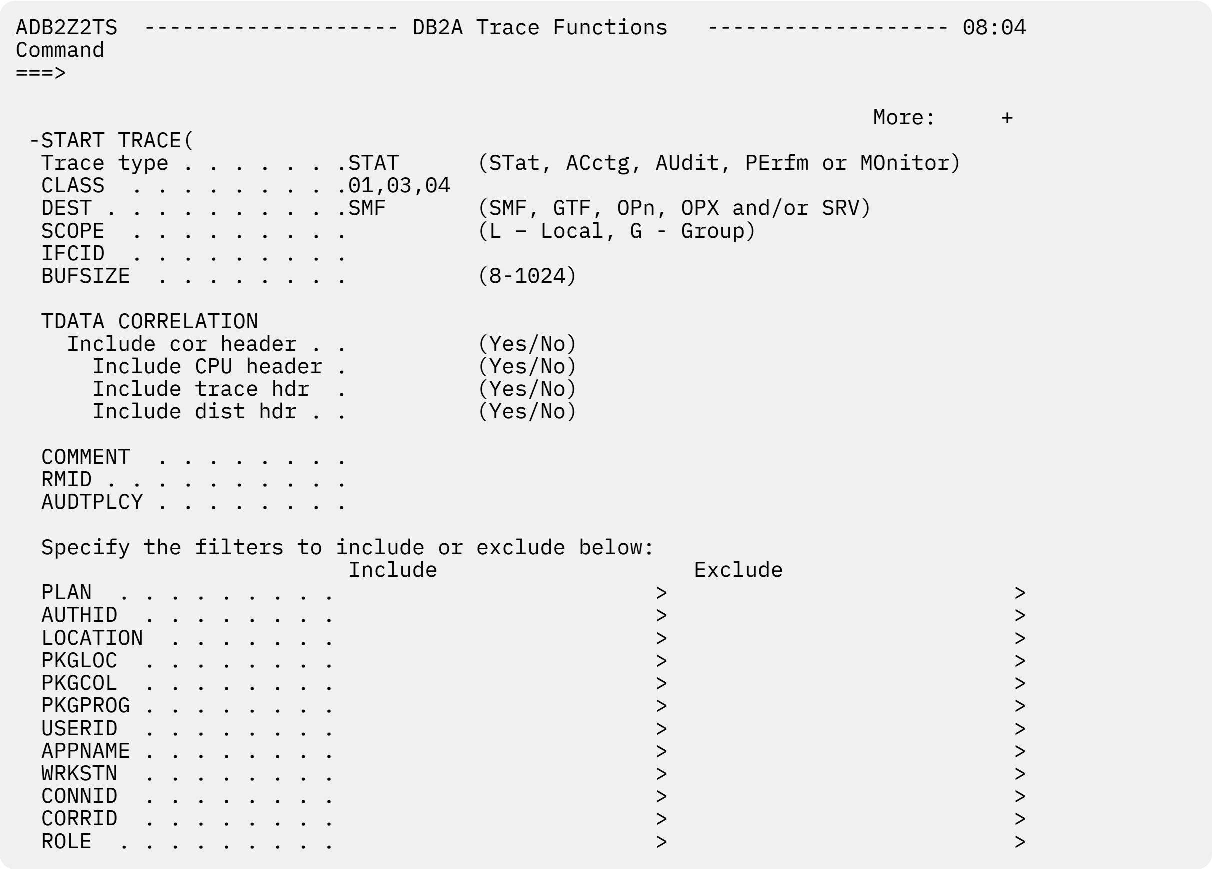
   IFCID
   The IFCIDs that are activated for the trace.

3. Issue one of the following line commands, and press Enter.

   DIS
   Displays all details for the last trace interval. (Db2 Admin Tool issues the Db2 command DISPLAY TRACE.)

   STA
   Starts the trace. (Db2 Admin Tool issues the Db2 command START TRACE.)

   If you specify STA, the **Trace Functions (ADB2Z2TS)** panel is displayed, and you can specify the filters for the trace options:



```
ADB2Z2TS  ------------------- DB2A Trace Functions   ------------------- 08:04
Command
===>

                                                                   More:     +
 -START TRACE(
  Trace type . . . . . . .STAT      (STat, ACctg, AUdit, PErfm or MOnitor)
  CLASS  . . . . . . . . .01,03,04
  DEST . . . . . . . . . .SMF       (SMF, GTF, OPn, OPX and/or SRV)
  SCOPE  . . . . . . . . .          (L - Local, G - Group)
  IFCID  . . . . . . . . .
  BUFSIZE  . . . . . . . .          (8-1024)

  TDATA CORRELATION
    Include cor header . .          (Yes/No)
      Include CPU header .          (Yes/No)
      Include trace hdr  .          (Yes/No)
      Include dist hdr . .          (Yes/No)

  COMMENT  . . . . . . . .
  RMID . . . . . . . . . .
  AUDTPLCY . . . . . . . .

  Specify the filters to include or exclude below:
                        Include                    Exclude
  PLAN  . . . . . . . . .                        >                           >
  AUTHID  . . . . . . . .                        >                           >
  LOCATION  . . . . . . .                        >                           >
  PKGLOC  . . . . . . . .                        >                           >
  PKGCOL  . . . . . . . .                        >                           >
  PKGPROG . . . . . . . .                        >                           >
  USERID  . . . . . . . .                        >                           >
  APPNAME . . . . . . . .                        >                           >
  WRKSTN  . . . . . . . .                        >                           >
  CONNID  . . . . . . . .                        >                           >
  CORRID  . . . . . . . .                        >                           >
  ROLE  . . . . . . . . .                        >                           >
```

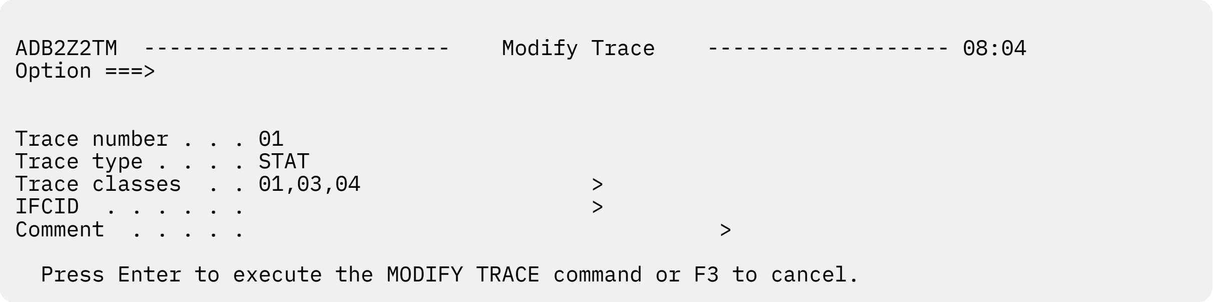
STO

Stops the trace. (Db2 Admin Tool issues the Db2 command STOP TRACE.)

MT

Changes the IFCIDs (trace events) that are associated with a particular active trace. (Db2 Admin Tool issues the Db2 command MODIFY TRACE.)

If you specify `MT`, the **Modify Trace (ADB2Z2TM)** panel is displayed and you can specify new values for **Trace classes**, **IFCID**, or **Comment**:



```
ADB2Z2TM  ------------------------    Modify Trace    ------------------- 08:04
Option ===>


Trace number . . . 01
Trace type . . . . STAT
Trace classes  . . 01,03,04                      >
IFCID  . . . . . .                               >
Comment  . . . . .                                         >

  Press Enter to execute the MODIFY TRACE command or F3 to cancel.
```

The information that Db2 Admin Tool returns from the commands is in ISPF browse format.

Related information

Db2 trace (Db2 12 for z/OS)

Displaying or updating the owner of resource limit (RLIMIT) ables

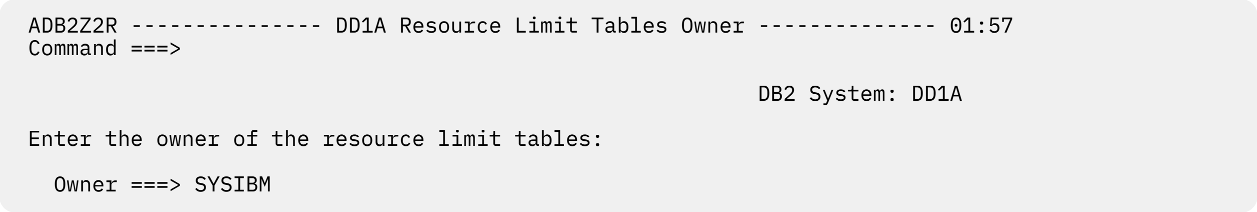
You can display or update the owner of the resource limit tables.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option 2R, and press Enter.

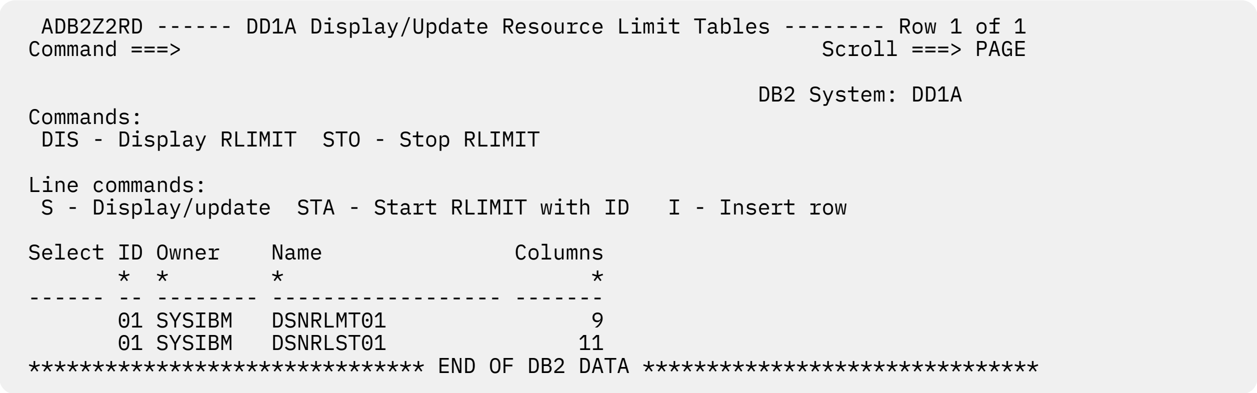
   The **Resource Limit Tables Owner (ADB2Z2R)** panel is displayed, as shown in the following figure.



```
ADB2Z2R --------------- DD1A Resource Limit Tables Owner -------------- 01:57
Command ===>

                                                         DB2 System: DD1A

Enter the owner of the resource limit tables:

  Owner ===> SYSIBM
```

*Figure 428. **Resource Limit Tables Owner (ADB2Z2R)** panel*

3. Specify the owner of the resource limit tables, and press Enter to display the resource limit tables owned by that owner, as shown in the following figure.



```
 ADB2Z2RD ------ DD1A Display/Update Resource Limit Tables -------- Row 1 of 1
Command ===>                                                Scroll ===> PAGE

                                                         DB2 System: DD1A
Commands:
 DIS - Display RLIMIT  STO - Stop RLIMIT

Line commands:
 S - Display/update  STA - Start RLIMIT with ID   I - Insert row

Select ID Owner    Name               Columns
       *  *        *                        *
------ -- -------- ------------------ -------
       01 SYSIBM   DSNRLMT01                9
       01 SYSIBM   DSNRLST01               11
******************************* END OF DB2 DATA *******************************
```

Figure 429. Display/Update Resource Limit (RLIMIT) Tables panel (ADB2Z2RD)

   The following fields are available on this panel:

   SELECT
   Input field where you enter one of the line commands listed on the panel.

   ID

   RLIMIT identifier.

   OWNER
   Authorization ID of the owner of the RLIMIT table.

   NAME
   Name of the RLIMIT table.

   COLUMNS
   Number of columns in the RLIMIT table.
4. Issue one of the following commands, and press Enter:

   DIS
   Displays the current status of the resource limit. This command is equivalent to the Db2 command DISPLAY RLIMIT.

The following figure shows the RLIMIT status information Db2 Admin Tool returns when you issue the DIS primary command.

```
 ADB2DB2O  ------- DD1A Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

  -STA RLIMIT ID=01


********************************* Top of Data **********************************
DSNT704I  #DSN9- SYSIBM.DSNRLST01 HAS BEEN STARTED FOR THE RESOURCE
LIMIT FACILITY
DSNT704I  #DSN9- SYSIBM.DSNRLMT01 HAS BEEN STARTED FOR THE RESOURCE
LIMIT FACILITY
DSN9022I  #DSN9- DSNTCSTR 'START RLIMIT' NORMAL COMPLETION
 ******************************** Bottom of Data ********************************
```

Figure 430. Display RLIMIT panel (ADB2DB2O)

STO

Stops the resource limit. This command is equivalent to the Db2 command STOP RLIMIT.

The following figure shows the information Db2 Admin Tool returns when you issue the STO primary command to stop the resource limit facility.

```
 ADB2DB2O ------- DD1A Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

  -STO RLIMIT ID=01


********************************* Top of Data **********************************
DSNT702I  #DSN9- RESOURCE LIMIT FACILITY HAS BEEN STOPPED. WAS USING
SYSIBM.DSNRLST01
DSNT702I  #DSN9- RESOURCE LIMIT FACILITY HAS BEEN STOPPED. WAS USING
SYSIBM.DSNRLMT01
DSN9022I  #DSN9- DSNTCSTP 'STOP RLIMIT' NORMAL COMPLETION
******************************** Bottom of Data ********************************
```

Figure 431. Stop RLIMIT panel (ADB2DB2O)

S

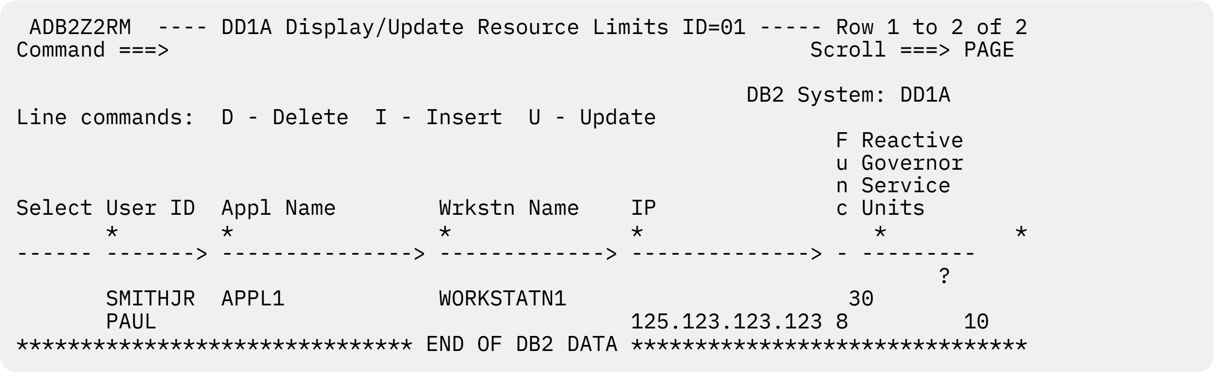
The S line command. It displays or updates the resource limit status and the resource limit status of resource limit table DSNRLMT*xx*.

The following figure shows the panel returned when you issued the S line command to show the content of the RLIMIT table and used the primary command PRE ON to show the predictive governor columns too.

```
 ADB2Z2RS ---- DD1A Display/Update Resource Limits ID=01 ---------- Row 1 of 1
 Command ===>                                                  Scroll ===> PAGE

                                                          DB2 System: DD1A
 Line commands:  D - Delete  I - Insert  U - Update
                                                               F    Reactive B
                                                               u     Govenor i
                                                               n     Service n
 Select Auth ID  Plan     Collection         Package  LU Name  c       Units d
        *        *        *                  *        *        *           * *
 ------ -------> -------- -----------------> -------> -------> - ----------- -
                                                                           ?
        VNDOKAV           COL1               PACK1    LU1      1           ? N
        VNDOKAV           XCOLL              XPACK    XLU      1           ?
        VNDOKAV           YCOLL              YPACK    YLU      1           ?
        VNDWLB1           WLBCOLLECTION      WLBPACKA WLBLU    1           ?
 ******************************* END OF DB2 DATA *******************************
```

Figure 432. Display RLIMIT panel (ADB2Z2RS)

The following figure shows the panel returned when you issue the S line command to show the column values of DSNRLMT*xx* resource table.



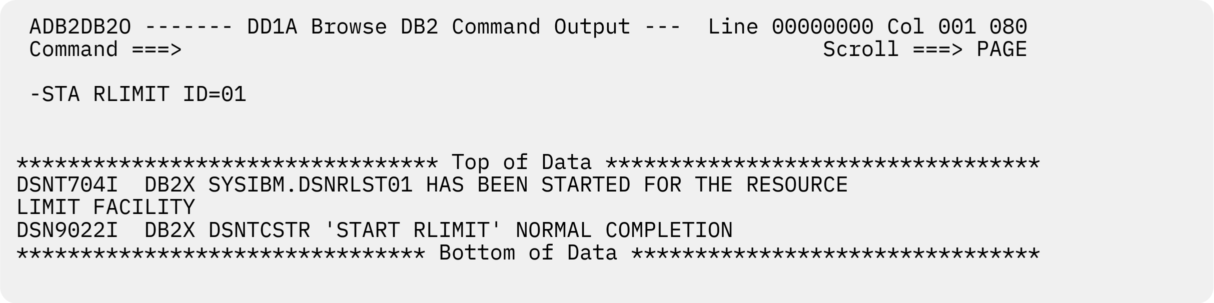
```
 ADB2Z2RM  ---- DD1A Display/Update Resource Limits ID=01 ----- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

                                                         DB2 System: DD1A
Line commands:  D - Delete  I - Insert  U - Update
                                                                F Reactive
                                                                u Governor
                                                                n Service
Select User ID  Appl Name        Wrkstn Name    IP              c Units
       *        *                *              *                  *          *
------ -------> ---------------> -------------> --------------> - ---------
                                                                        ?
       SMITHJR  APPL1            WORKSTATN1                      30
       PAUL                                     125.123.123.123 8        10
******************************* END OF DB2 DATA *******************************
```

Figure 433. Display RLIMIT panel (ADB2Z2RM)

STA

Starts the resource limit with ID.

The following figure shows the information Db2 Admin Tool returns when you issue the STA line command to start the resource limit facility with a particular ID.



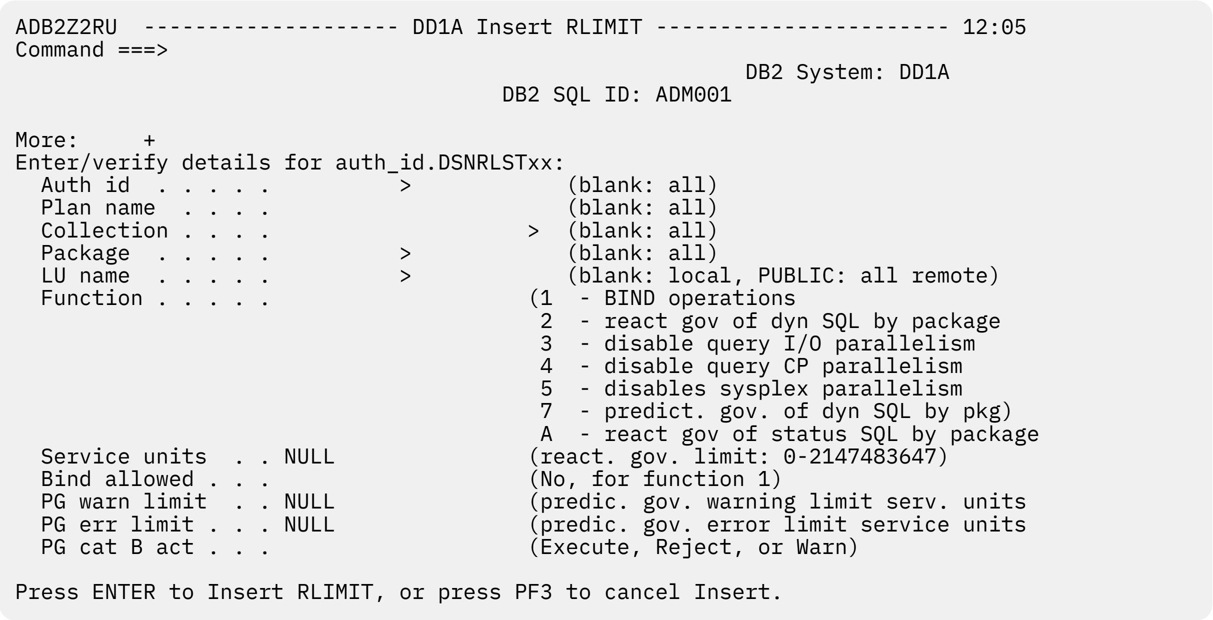
```
 ADB2DB2O ------- DD1A Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

 -STA RLIMIT ID=01


********************************* Top of Data **********************************
DSNT704I  DB2X SYSIBM.DSNRLST01 HAS BEEN STARTED FOR THE RESOURCE
LIMIT FACILITY
DSN9022I  DB2X DSNTCSTR 'START RLIMIT' NORMAL COMPLETION
******************************** Bottom of Data ********************************
```

Figure 434. Start RLIMIT panel (ADB2DB2O)

I

Inserts rows into the resource limit table and inserts or updates column values for the DSNRLMT*xx* resource limit table.

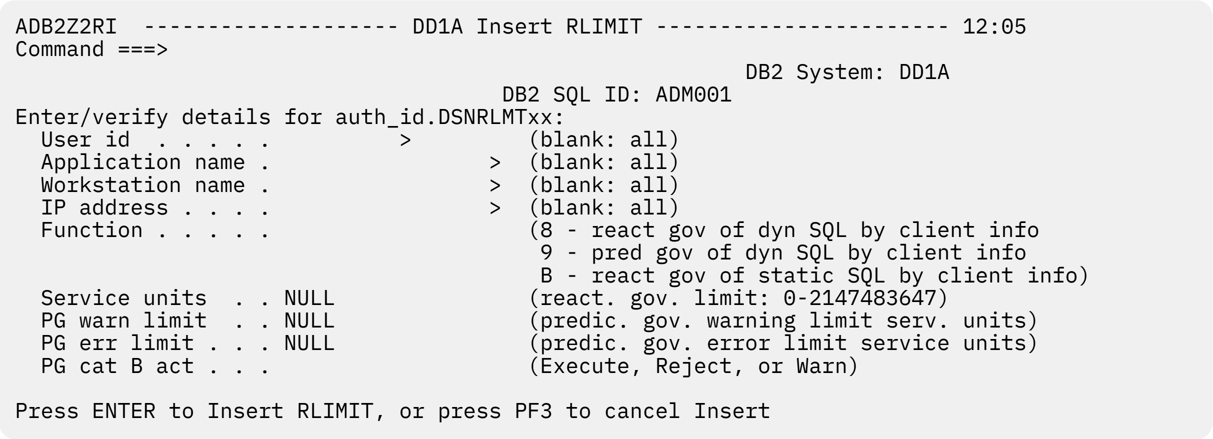
The following figure shows the output when you enter the I line command in front of a row from the RLIMIT table in Figure 432 on page 908. On the Insert RLIMIT panel, as shown in the following figure, you can enter values for a new row in the RLIMIT table.



```
ADB2Z2RU  -------------------- DD1A Insert RLIMIT ---------------------- 12:05
Command ===>
                                                         DB2 System: DD1A
                                      DB2 SQL ID: ADM001

More:     +
Enter/verify details for auth_id.DSNRLSTxx:
  Auth id  . . . . .          >            (blank: all)
  Plan name  . . . .                       (blank: all)
  Collection . . . .                    >  (blank: all)
  Package  . . . . .          >            (blank: all)
  LU name  . . . . .          >            (blank: local, PUBLIC: all remote)
  Function . . . . .                    (1  - BIND operations
                                         2  - react gov of dyn SQL by package
                                         3  - disable query I/O parallelism
                                         4  - disable query CP parallelism
                                         5  - disables sysplex parallelism
                                         7  - predict. gov. of dyn SQL by pkg)
                                         A  - react gov of status SQL by package
  Service units  . . NULL               (react. gov. limit: 0-2147483647)
  Bind allowed . . .                    (No, for function 1)
  PG warn limit  . . NULL               (predic. gov. warning limit serv. units
  PG err limit . . . NULL               (predic. gov. error limit service units
  PG cat B act . . .                    (Execute, Reject, or Warn)

Press ENTER to Insert RLIMIT, or press PF3 to cancel Insert.
```

Figure 435. Insert RLIMIT panel (ADB2Z2RU)

The following figure shows the output when you enter the I line command in front of a row from the DSNRLMT*xx* RLIMIT table in panel ADB2ZRM.



```
ADB2Z2RI  -------------------- DD1A Insert RLIMIT ---------------------- 12:05
Command ===>
                                                         DB2 System: DD1A
                                      DB2 SQL ID: ADM001
Enter/verify details for auth_id.DSNRLMTxx:
  User id  . . . . .          >         (blank: all)
  Application name .               >    (blank: all)
  Workstation name .               >    (blank: all)
  IP address . . . .               >    (blank: all)
  Function . . . . .                    (8 - react gov of dyn SQL by client info
                                         9 - pred gov of dyn SQL by client info
                                         B - react gov of static SQL by client info)
  Service units  . . NULL               (react. gov. limit: 0-2147483647)
  PG warn limit  . . NULL               (predic. gov. warning limit serv. units)
  PG err limit . . . NULL               (predic. gov. error limit service units)
  PG cat B act . . .                    (Execute, Reject, or Warn)

Press ENTER to Insert RLIMIT, or press PF3 to cancel Insert
```

Figure 436. Insert RLIMIT panel (ADB2Z2RI)

The information that Db2 Admin Tool returns to you from the commands is in ISPF browse format.

Stopping Db2

This procedure explains how to stop the Db2 subsystem.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option 2S, and press Enter.

   The **Stop DB2 (ADB2Z2S)** panel is displayed, as shown in the following figure.

```
 DB2 Admin ------------------------ DB2X Stop DB2 ------------------------ 16:07
 Command ===>

  -STOP DB2

   MODE(
   Stop mode          ===>          (Quiesce or Force, default is quiesce)
 )
 Note: After using FORCE mode, exit from DB2 Admin without issuing any further
 SQL statements.
```

*Figure 437. **Stop DB2 (ADB2Z2S)** panel*

3. In the **Stop mode** field, specify `Quiesce` or `Force`, and press Enter.

   Db2 Admin Tool accomplishes this task by issuing the Db2 command STOP DB2.

   The information that Db2 Admin Tool returns to you from the command is in ISPF browse format.

Displaying group information

You can display information about the data sharing group to which a Db2 subsystem belongs.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option 2G, and press Enter.

   The **Display Group (ADB2DB2O)** panel is displayed, as shown in the following figure.

```
 DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

 -DIS GROUP


********************************* Top of Data **********************************
DB2X100I -DB81 DB2XGCMD
*** BEGIN DISPLAY OF GROUP(DSNDB26 ) GROUP LEVEL(810)
                                     GROUP ATTACH NAME(DB26)
--------------------------------------------------------------------
DB2                                    DB2 SYSTEM    IRLM
MEMBER   ID  SUBSYS CMDPREF   STATUS   LVL NAME      SUBSYS IRLMPROC
-------- --- ----   --------  -------- --- --------  ----   --------
DB81       1 DB81   -DB81     ACTIVE   810 ZPLEX     IR81   DB81IRLM
DB82       2 DB82   -DB82     FAILED   810 ZPLEX1    IR82   DB82IRLM
--------------------------------------------------------------------
SCA   STRUCTURE SIZE:     4096 KB, STATUS= AC,   SCA IN USE:     2 %
LOCK1 STRUCTURE SIZE:     4096 KB,
NUMBER  LOCK ENTRIES:     1048576
NUMBER  LIST ENTRIES:       13878, LIST ENTRIES  IN USE:          22
*** END DISPLAY OF GROUP(DSNDB26 )
DSN9022I -DB81 DB2XGCMD 'DISPLAY GROUP ' NORMAL COMPLETION
******************************** Bottom of Data ********************************
```

*Figure 438. **Display Group (ADB2DB2O)** panel*

Db2 Admin Tool generates this panel by issuing the Db2 DISPLAY GROUP command.

Displaying or managing batch checkpoint tables

The Db2 Admin Tool Batch Restart program, ADBTEP2, provides the ability to restart or resume the execution of an input stream of SQL statements, utilities, and Db2 commands in a batch job at an intermediate point, in the event that any one of the statements in that input stream should fail.

About this task

The information to monitor the execution of the input stream is stored in a Db2 table referred to as the checkpoint table.

The **Manage Batch Job Checkpoint Table (ADB2Z2B)** panel allows you to display and manage the checkpoint table for batch jobs running ADBTEP2. A row exists in the checkpoint table for each active and abnormally terminated job running ADBTEP2.

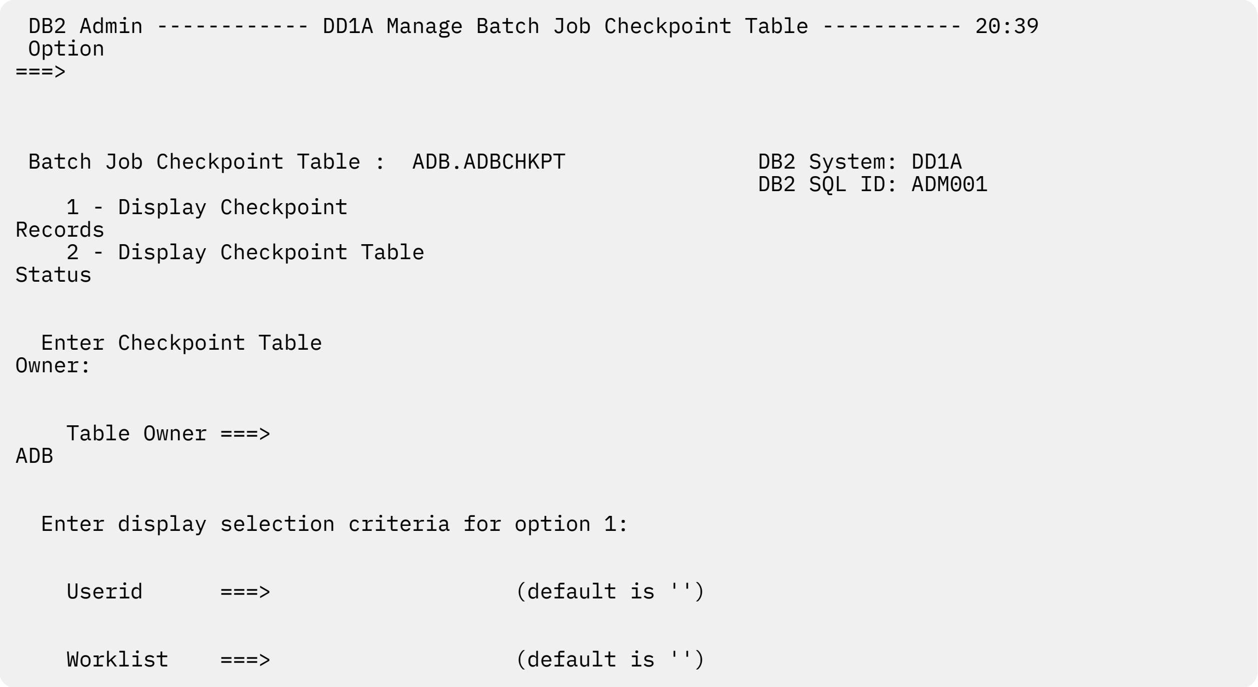
To display and manage the checkpoint table for the batch jobs that running ADBTEP2:

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option 2B, and press Enter.

   The **Manage Batch Job Checkpoint Table (ADB2Z2B)** panel is displayed, as shown in the following figure.



```
 DB2 Admin ------------ DD1A Manage Batch Job Checkpoint Table ----------- 20:39
 Option
===>



 Batch Job Checkpoint Table :  ADB.ADBCHKPT               DB2 System: DD1A
                                                          DB2 SQL ID: ADM001
    1 - Display Checkpoint
Records
    2 - Display Checkpoint Table
Status


  Enter Checkpoint Table
Owner:


    Table Owner ===>
ADB


  Enter display selection criteria for option 1:


    Userid      ===>                  (default is '')


    Worklist    ===>                  (default is '')
```

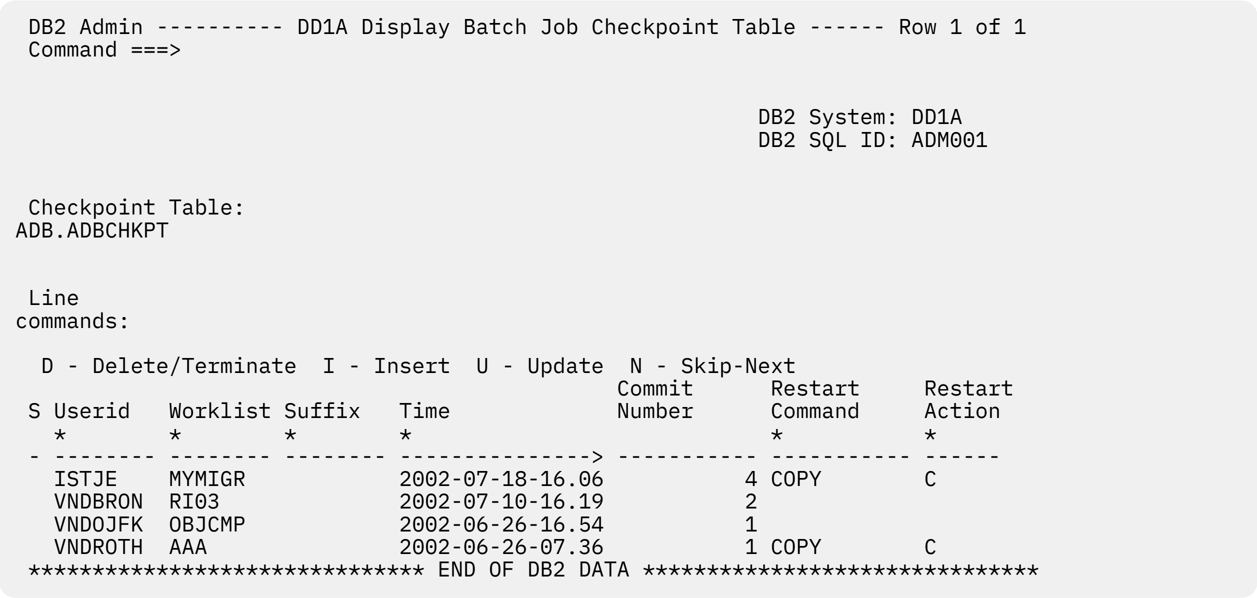
*Figure 439. **Manage Batch Job Checkpoint Table (ADB2Z2B)** panel*

3. Select one of the following options and press Enter.

   1 - Display Checkpoint Records
   Displays all checkpoint records. Use option 1 to terminate an active ADBTEP2 job, update or delete a record of an abnormal terminated job, or insert a new checkpoint record.

   2 - Display Checkpoint Table Status
   Displays information about the checkpoint table. Use this option to issue any request against the checkpoint table that is supported by Db2 Admin Tool, such as GRANT or REVOKE.

   The following figure shows the rows in the table you selected.



```
 DB2 Admin ---------- DD1A Display Batch Job Checkpoint Table ------ Row 1 of 1
 Command ===>


                                                          DB2 System: DD1A
                                                          DB2 SQL ID: ADM001


 Checkpoint Table:
ADB.ADBCHKPT


 Line
commands:

  D - Delete/Terminate  I - Insert  U - Update  N - Skip-Next
                                               Commit      Restart     Restart
 S Userid   Worklist Suffix   Time             Number      Command     Action
   *        *        *        *                            *           *
 - -------- -------- -------- ---------------> ----------- ----------- ------
   ISTJE    MYMIGR            2002-07-18-16.06           4 COPY        C
   VNDBRON  RI03              2002-07-10-16.19           2
   VNDOJFK  OBJCMP            2002-06-26-16.54           1
   VNDROTH  AAA               2002-06-26-07.36           1 COPY        C
 ******************************* END OF DB2 DATA *******************************
```

Figure 440. Display Batch Job Checkpoint Table panel (ADB2Z2B1)

When data is unloaded in one job and is then reloaded in another job, the unload suffix has the following format: U*xxxxx*. The corresponding reload is R*xxxxx*. An additional suffix might also exist, in the format @*xxxxx*. Never attempt to update or modify the @*xxxxx* record. Delete this record only if you are abandoning a current run of a work statement list. The @*xxxxx* record is deleted by the job using R*xxxxx*.

Use the following line commands to change the content of the table:

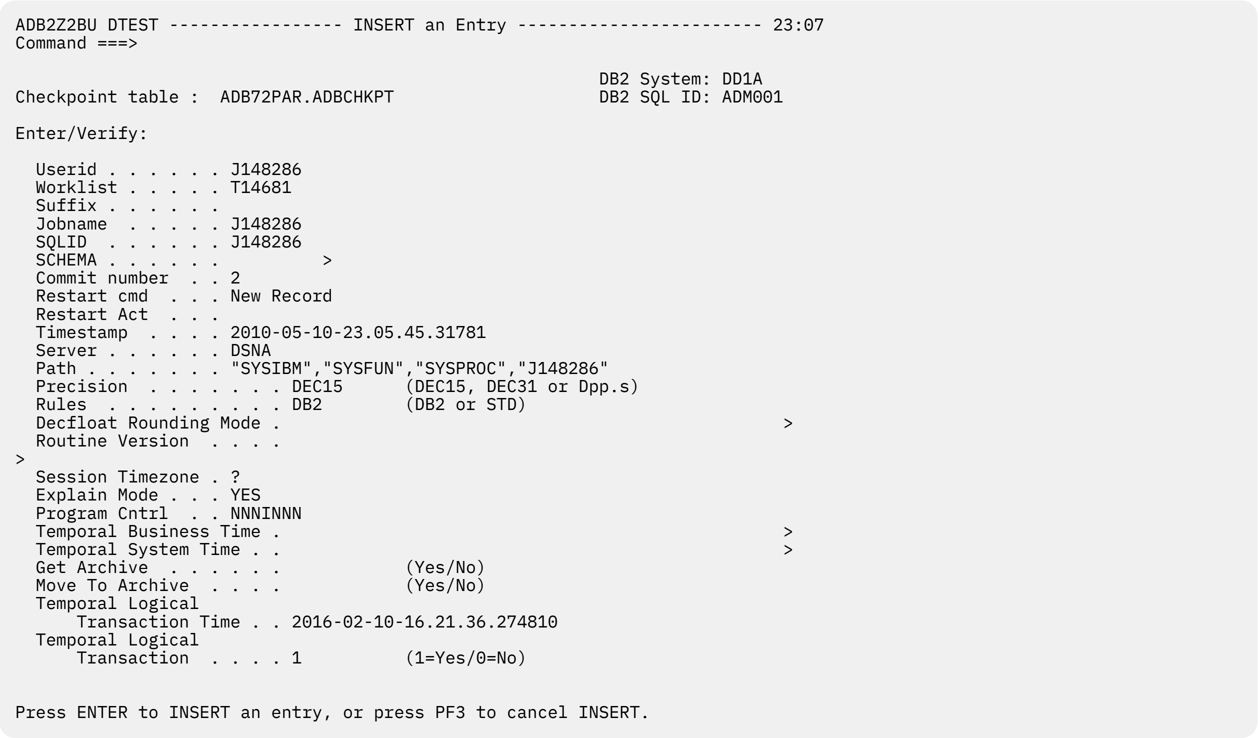
D
: To DELETE the row of an abnormally terminated job or to terminate an active job.

I
: To INSERT a new row. Row values can be entered on the next panel displayed.

U
: To UPDATE the row of an abnormally terminated job. If the job is executing, the request is rejected. Row values can be changed on the next panel.

N
: To instruct ADBTEP2 to skip to the next commit instruction.

4. If you use the I or U line commands, the insert or update a checkpoint record panel (ADB2Z2BU) displays. The schema and sqlid values will be used during a restart for setting the current sqlid and current schema special registers at the point of restart. When you update a checkpoint record that does not have a SCHEMA value (is null), the panel value displayed will be blank . If you do not enter a new value, the SCHEMA value remains null. When you insert a new checkpoint record using the panels, if you do not enter a non-blank value, a null value will be stored.



```
ADB2Z2BU DTEST ----------------- INSERT an Entry ------------------------ 23:07
Command ===>

                                                         DB2 System: DD1A
Checkpoint table :  ADB72PAR.ADBCHKPT                    DB2 SQL ID: ADM001

Enter/Verify:

  Userid . . . . . . J148286
  Worklist . . . . . T14681
  Suffix . . . . . .
  Jobname  . . . . . J148286
  SQLID  . . . . . . J148286
  SCHEMA . . . . . .          >
  Commit number  . . 2
  Restart cmd  . . . New Record
  Restart Act  . . .
  Timestamp  . . . . 2010-05-10-23.05.45.31781
  Server . . . . . . DSNA
  Path . . . . . . . "SYSIBM","SYSFUN","SYSPROC","J148286"
  Precision  . . . . . . . DEC15      (DEC15, DEC31 or Dpp.s)
  Rules  . . . . . . . . . DB2        (DB2 or STD)
  Decfloat Rounding Mode .                                          >
  Routine Version  . . . .
>
  Session Timezone . ?
  Explain Mode . . . YES
  Program Cntrl  . . NNNINNN
  Temporal Business Time .                                          >
  Temporal System Time . .                                          >
  Get Archive  . . . . . .            (Yes/No)
  Move To Archive  . . . .            (Yes/No)
  Temporal Logical
      Transaction Time . . 2016-02-10-16.21.36.274810
  Temporal Logical
      Transaction  . . . . 1          (1=Yes/0=No)


Press ENTER to INSERT an entry, or press PF3 to cancel INSERT.
```

Figure 441. Insert or update a checkpoint record panel (ADB2Z2BU)

Managing audit policies

A Db2 *audit policy* is a set of criteria that determines the categories to be audited.

About this task

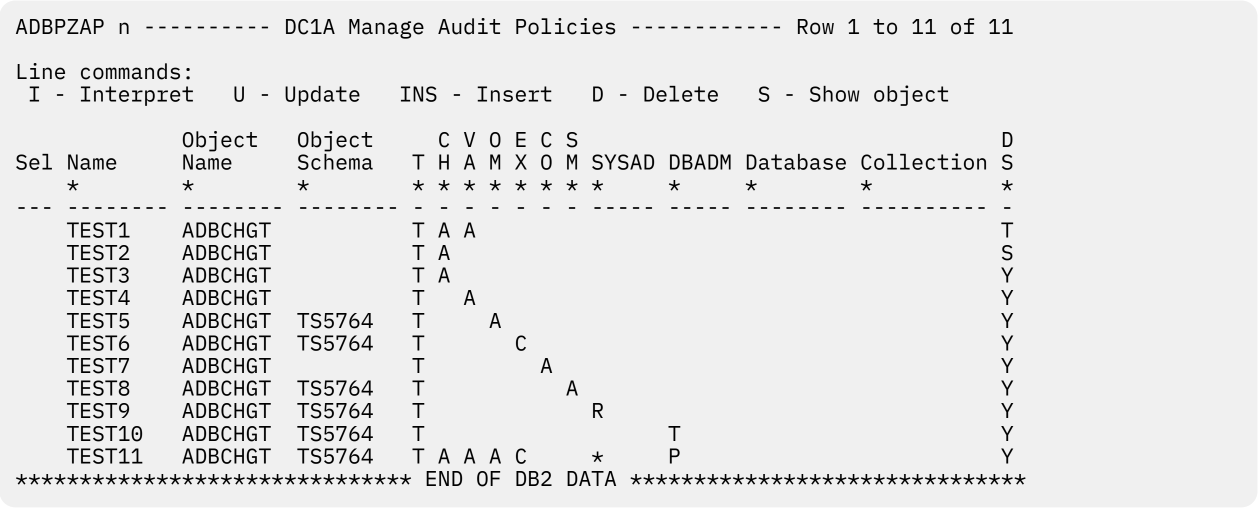
You create an audit policy by inserting a row into the Db2 catalog table SYSIBM.SYSAUDITPOLICIES. You can then edit and delete these policies by modifying the rows in this table. Db2 Admin Tool helps you perform these steps.

Procedure

To manage audit policies:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AP, and press Enter.

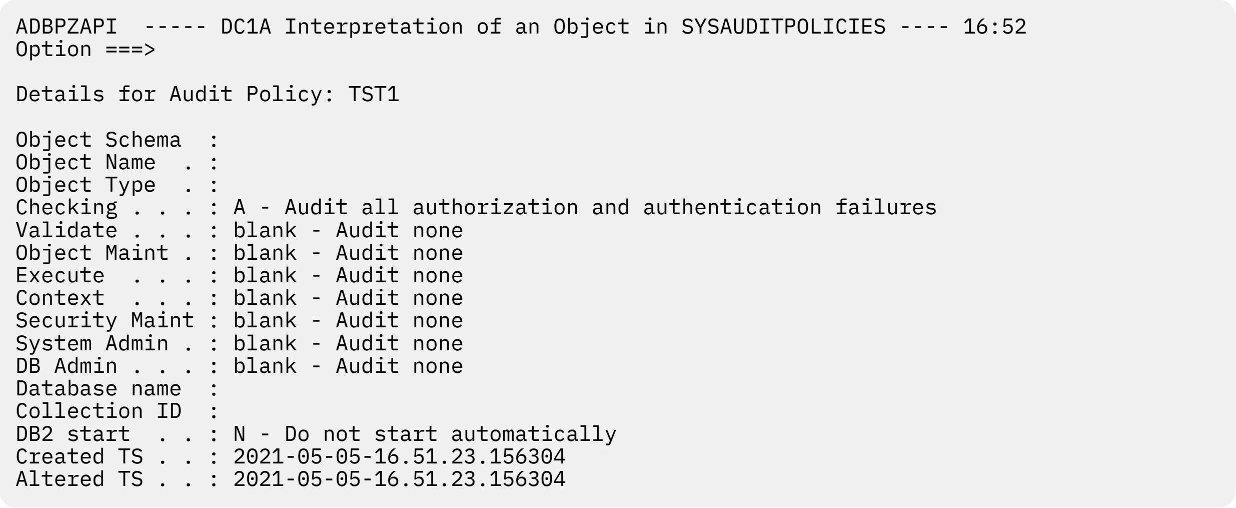
   The **Manage Audit Policies (ADBPZAP)** panel displays the audit policies that are stored in SYSAUDITPOLICIES:



```
ADBPZAP n ---------- DC1A Manage Audit Policies ------------ Row 1 to 11 of 11

Line commands:
 I - Interpret   U - Update   INS - Insert   D - Delete   S - Show object

             Object   Object     C V O E C S                         D
Sel Name     Name     Schema   T H A M X O M SYSAD DBADM Database Collection S
    *        *        *        * * * * * * * *     *     *        *          *
--- -------- -------- -------- - - - - - - - ----- ----- -------- ---------- -
    TEST1    ADBCHGT           T A A                                         T
    TEST2    ADBCHGT           T A                                           S
    TEST3    ADBCHGT           T A                                           Y
    TEST4    ADBCHGT           T   A                                         Y
    TEST5    ADBCHGT  TS5764   T     A                                       Y
    TEST6    ADBCHGT  TS5764   T       C                                     Y
    TEST7    ADBCHGT           T         A                                   Y
    TEST8    ADBCHGT  TS5764   T           A                                 Y
    TEST9    ADBCHGT  TS5764   T             R                               Y
    TEST10   ADBCHGT  TS5764   T                   T                         Y
    TEST11   ADBCHGT  TS5764   T A A A C     *     P                         Y
******************************* END OF DB2 DATA *******************************
```

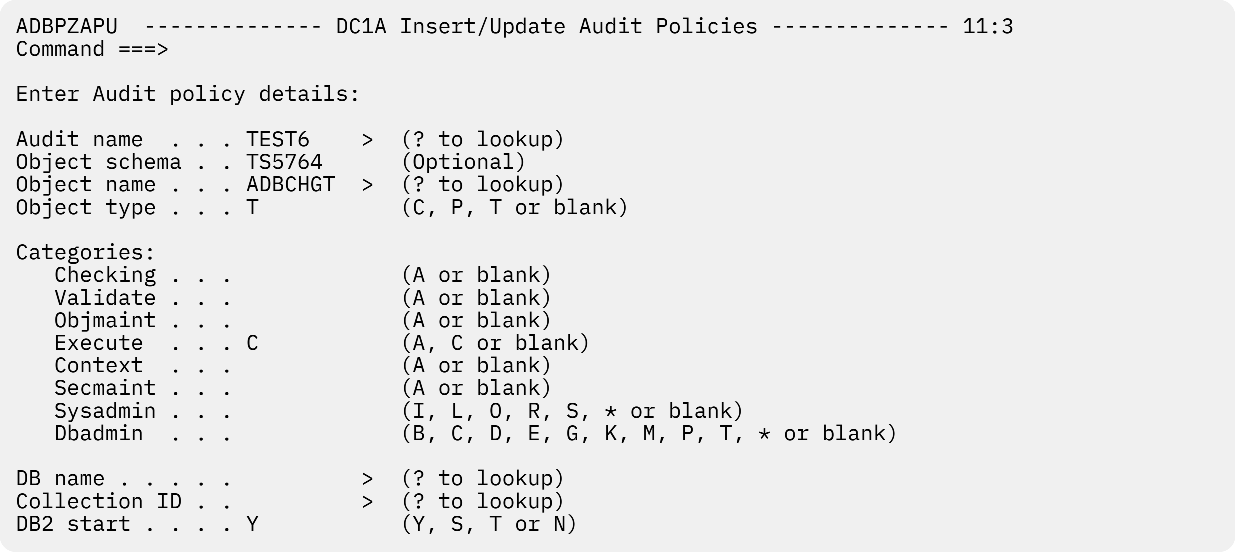
3. Use the line commands on the **Manage Audit Policies (ADBPZAP)** panel to view, add, and update any audit policies as needed:

   - If you view a policy (by using the I line command), the **Interpretation of an Object in SYSAUDITPOLICIES (ADBPZAPI)** panel displays the policy details:



```
ADBPZAPI  ----- DC1A Interpretation of an Object in SYSAUDITPOLICIES ---- 16:52
Option ===>

Details for Audit Policy: TST1

Object Schema  :
Object Name  . :
Object Type  . :
Checking . . . : A - Audit all authorization and authentication failures
Validate . . . : blank - Audit none
Object Maint . : blank - Audit none
Execute  . . . : blank - Audit none
Context  . . . : blank - Audit none
Security Maint : blank - Audit none
System Admin . : blank - Audit none
DB Admin . . . : blank - Audit none
Database name  :
Collection ID  :
DB2 start  . . : N - Do not start automatically
Created TS . . : 2021-05-05-16.51.23.156304
Altered TS . . : 2021-05-05-16.51.23.156304
```

   - If you insert a new policy (with the INS line command) or update a policy (with the U line command), the **Insert/Update Audit Policies (ADBPZAPU)** panel is displayed:



```
ADBPZAPU  -------------- DC1A Insert/Update Audit Policies -------------- 11:3
Command ===>

Enter Audit policy details:

Audit name  . . . TEST6    >  (? to lookup)
Object schema . . TS5764      (Optional)
Object name . . . ADBCHGT  >  (? to lookup)
Object type . . . T           (C, P, T or blank)

Categories:
   Checking . . .             (A or blank)
   Validate . . .             (A or blank)
   Objmaint . . .             (A or blank)
   Execute  . . . C           (A, C or blank)
   Context  . . .             (A or blank)
   Secmaint . . .             (A or blank)
   Sysadmin . . .             (I, L, O, R, S, * or blank)
   Dbadmin  . . .             (B, C, D, E, G, K, M, P, T, * or blank)

DB name . . . . .          >  (? to lookup)
Collection ID . .          >  (? to lookup)
DB2 start . . . . Y           (Y, S, T or N)
```

On this panel, enter the values that you want inserted or updated in the SYSAUDITPOLICIES table and press Enter.

Tip: For Db2 12 function level 509 or higher, you can create a tamper-proof audit policy, which requires special authorization to modify or stop. To create such a policy, specify T in the **DB2 start** field.

Related information

Db2 audit policy (Db2 12 for z/OS documentation)

Package management

Packages are produced during the Db2 program preparation process when you bind a DBRM. You can use Db2 Admin Tool to perform various administrative tasks for your packages, such as rebinding a package or viewing the SQL statements that are associated with a package.

Related reference

Preparation process for an application program (Db2 12 for z/OS)

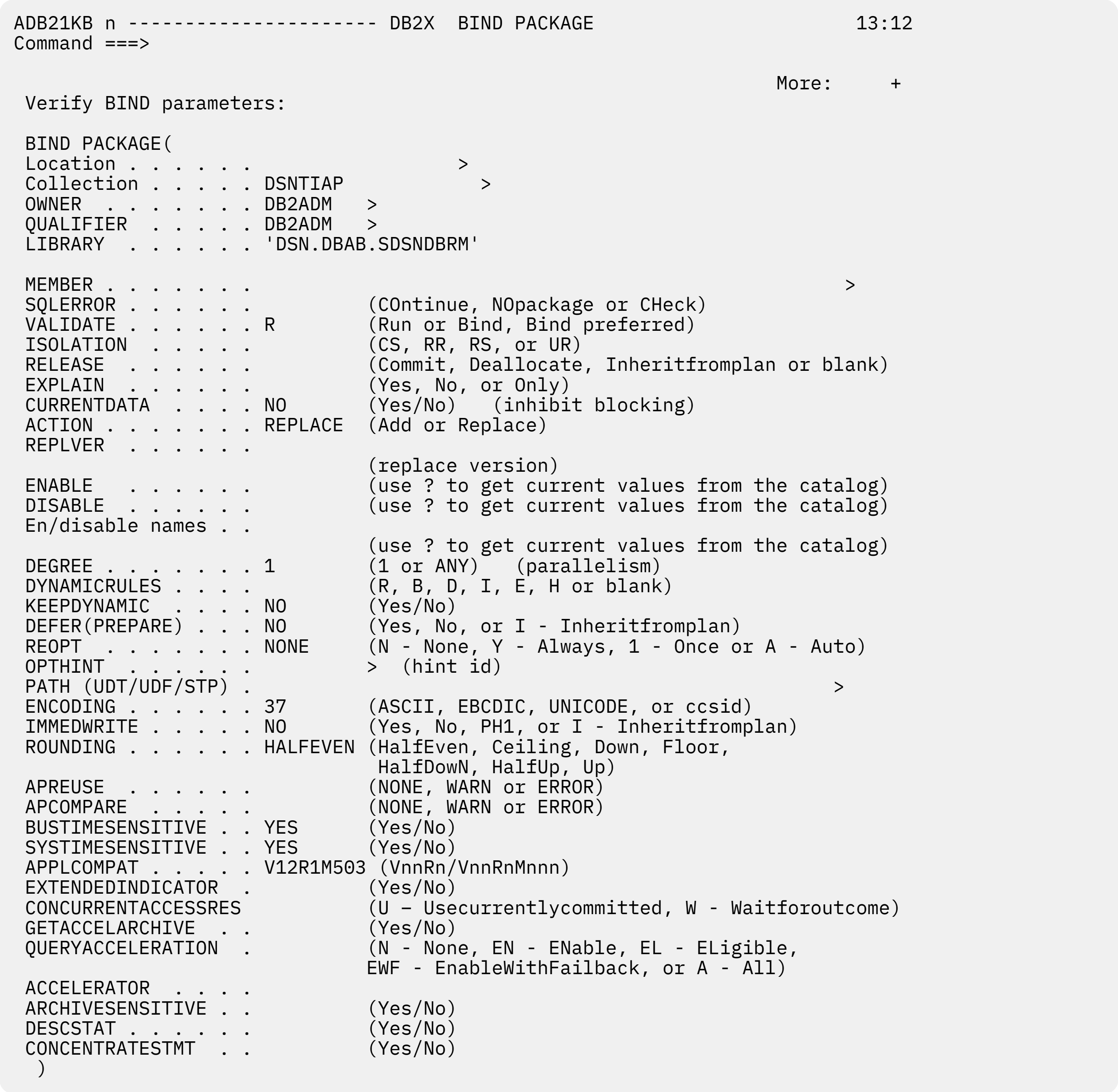
Binding packages and generating BIND statements

If you changed the SQL in your application, you need to bind the associated package to replace the existing one. You might also want to generate BIND statements for existing packages and save those statements without executing them.

Procedure

To bind an existing package or generate a BIND statement:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option K (for packages), and press Enter.
3. On the **Packages (ADB21K)** panel, specify the B line command next to the package that you want to bind.
4. On the **Bind package (ADB21KB)** panel, specify the options that you want to build an application package, and press Enter:



```
ADB21KB n ---------------------- DB2X  BIND PACKAGE                       13:12
Command ===>
                                                                   More:     +
 Verify BIND parameters:

 BIND PACKAGE(
 Location . . . . . .                  >
 Collection . . . . . DSNTIAP            >
 OWNER  . . . . . . . DB2ADM   >
 QUALIFIER  . . . . . DB2ADM   >
 LIBRARY  . . . . . . 'DSN.DBAB.SDSNDBRM'

 MEMBER . . . . . . .                                                    >
 SQLERROR . . . . . .          (COntinue, NOpackage or CHeck)
 VALIDATE . . . . . . R        (Run or Bind, Bind preferred)
 ISOLATION  . . . . .          (CS, RR, RS, or UR)
 RELEASE  . . . . . .          (Commit, Deallocate, Inheritfromplan or blank)
 EXPLAIN  . . . . . .          (Yes, No, or Only)
 CURRENTDATA  . . . . NO       (Yes/No)   (inhibit blocking)
 ACTION . . . . . . . REPLACE  (Add or Replace)
 REPLVER  . . . . . .
                               (replace version)
 ENABLE   . . . . . .          (use ? to get current values from the catalog)
 DISABLE  . . . . . .          (use ? to get current values from the catalog)
 En/disable names . .
                               (use ? to get current values from the catalog)
 DEGREE . . . . . . . 1        (1 or ANY)   (parallelism)
 DYNAMICRULES . . . .          (R, B, D, I, E, H or blank)
 KEEPDYNAMIC  . . . . NO       (Yes/No)
 DEFER(PREPARE) . . . NO       (Yes, No, or I - Inheritfromplan)
 REOPT  . . . . . . . NONE     (N - None, Y - Always, 1 - Once or A - Auto)
 OPTHINT  . . . . . .          >  (hint id)
 PATH (UDT/UDF/STP) .                                                   >
 ENCODING . . . . . . 37       (ASCII, EBCDIC, UNICODE, or ccsid)
 IMMEDWRITE . . . . . NO       (Yes, No, PH1, or I - Inheritfromplan)
 ROUNDING . . . . . . HALFEVEN (HalfEven, Ceiling, Down, Floor,
                                HalfDownN, HalfUp, Up)
 APREUSE  . . . . . .          (NONE, WARN or ERROR)
 APCOMPARE  . . . . .          (NONE, WARN or ERROR)
 BUSTIMESENSITIVE . . YES      (Yes/No)
 SYSTIMESENSITIVE . . YES      (Yes/No)
 APPLCOMPAT . . . . . V12R1M503 (VnnRn/VnnRnMnnn)
 EXTENDEDINDICATOR  .          (Yes/No)
 CONCURRENTACCESSRES           (U – Usecurrentlycommitted, W - Waitforoutcome)
 GETACCELARCHIVE  . .          (Yes/No)
 QUERYACCELERATION  .          (N - None, EN - ENable, EL - ELigible,
                               EWF - EnableWithFailback, or A - All)
 ACCELERATOR  . . . .
 ARCHIVESENSITIVE . .          (Yes/No)
 DESCSTAT . . . . . .          (Yes/No)
 CONCENTRATESTMT  . .          (Yes/No)
  )
```

Figure 442. ***Bind package (ADB21KB)*** *panel*

After you press Enter, either the BIND command is executed, or the **Statement Execution Prompt (ADB2PSTM)** panel is displayed where you can select what you want to do with the generated BIND statement. (Whether this panel is displayed depends on your prompt options; see "Changing Db2 Admin Tool prompt options" on page 243.) If you want to save the BIND statement, specify option 3 on the **Statement Execution Prompt (ADB2PSTM)** panel.

Rebinding packages

You need to rebind a package when you make changes that affect the package, but the SQL statements in the package have not changed. For example, if an associated object changed or an index was created, you need to rebind the package. When you rebind a package, the Db2 optimizer evaluates whether any access path changes are needed.

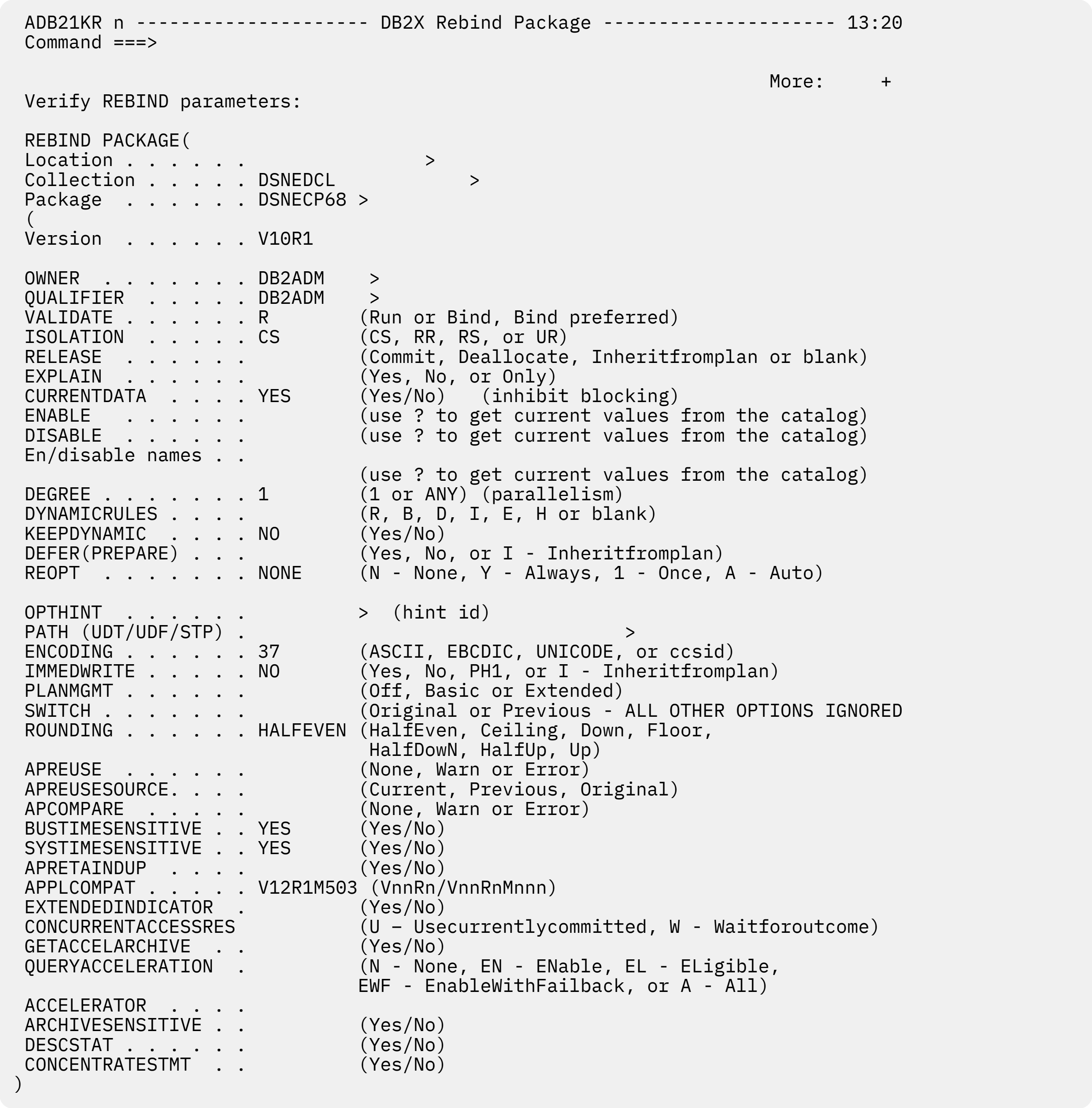
Procedure

To rebind a package:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.

2. On the **System Catalog (ADB21)** panel, specify option K (for packages), and press Enter.
3. On the **Packages (ADB21K)** panel, specify the RB line command next to the package that you want to rebind.
4. On the **Rebind package (ADB21KR)** panel, specify the options that you want, and press Enter.

   If you are changing the owner or qualifier for the package, set the PLANMGMT parameter to OFF or leave it blank.



```
 ADB21KR n --------------------- DB2X Rebind Package --------------------- 13:20
 Command ===>

                                                                    More:     +
 Verify REBIND parameters:

 REBIND PACKAGE(
 Location . . . . . .                >
 Collection . . . . . DSNEDCL            >
 Package  . . . . . . DSNECP68 >
 (
 Version  . . . . . . V10R1

 OWNER  . . . . . . . DB2ADM    >
 QUALIFIER  . . . . . DB2ADM    >
 VALIDATE . . . . . . R        (Run or Bind, Bind preferred)
 ISOLATION  . . . . . CS       (CS, RR, RS, or UR)
 RELEASE  . . . . . .          (Commit, Deallocate, Inheritfromplan or blank)
 EXPLAIN  . . . . . .          (Yes, No, or Only)
 CURRENTDATA  . . . . YES      (Yes/No)   (inhibit blocking)
 ENABLE   . . . . . .          (use ? to get current values from the catalog)
 DISABLE  . . . . . .          (use ? to get current values from the catalog)
 En/disable names . .
                               (use ? to get current values from the catalog)
 DEGREE . . . . . . . 1        (1 or ANY) (parallelism)
 DYNAMICRULES . . . .          (R, B, D, I, E, H or blank)
 KEEPDYNAMIC  . . . . NO       (Yes/No)
 DEFER(PREPARE) . . .          (Yes, No, or I - Inheritfromplan)
 REOPT  . . . . . . . NONE     (N - None, Y - Always, 1 - Once, A - Auto)

 OPTHINT  . . . . . .          >  (hint id)
 PATH (UDT/UDF/STP) .                                    >
 ENCODING . . . . . . 37       (ASCII, EBCDIC, UNICODE, or ccsid)
 IMMEDWRITE . . . . . NO       (Yes, No, PH1, or I - Inheritfromplan)
 PLANMGMT . . . . . .          (Off, Basic or Extended)
 SWITCH . . . . . . .          (Original or Previous - ALL OTHER OPTIONS IGNORED
 ROUNDING . . . . . . HALFEVEN (HalfEven, Ceiling, Down, Floor,
                                HalfDowN, HalfUp, Up)
 APREUSE  . . . . . .          (None, Warn or Error)
 APREUSESOURCE. . . .          (Current, Previous, Original)
 APCOMPARE  . . . . .          (None, Warn or Error)
 BUSTIMESENSITIVE . . YES      (Yes/No)
 SYSTIMESENSITIVE . . YES      (Yes/No)
 APRETAINDUP  . . . .          (Yes/No)
 APPLCOMPAT . . . . . V12R1M503 (VnnRn/VnnRnMnnn)
 EXTENDEDINDICATOR  .          (Yes/No)
 CONCURRENTACCESSRES           (U – Usecurrentlycommitted, W - Waitforoutcome)
 GETACCELARCHIVE  . .          (Yes/No)
 QUERYACCELERATION  .          (N - None, EN - ENable, EL - ELigible,
                               EWF - EnableWithFailback, or A - All)
 ACCELERATOR  . . . .
 ARCHIVESENSITIVE . .          (Yes/No)
 DESCSTAT . . . . . .          (Yes/No)
 CONCENTRATESTMT  . .          (Yes/No)
)
```

Figure 443. ***Rebind package (ADB21KR)*** *panel*

Freeing packages

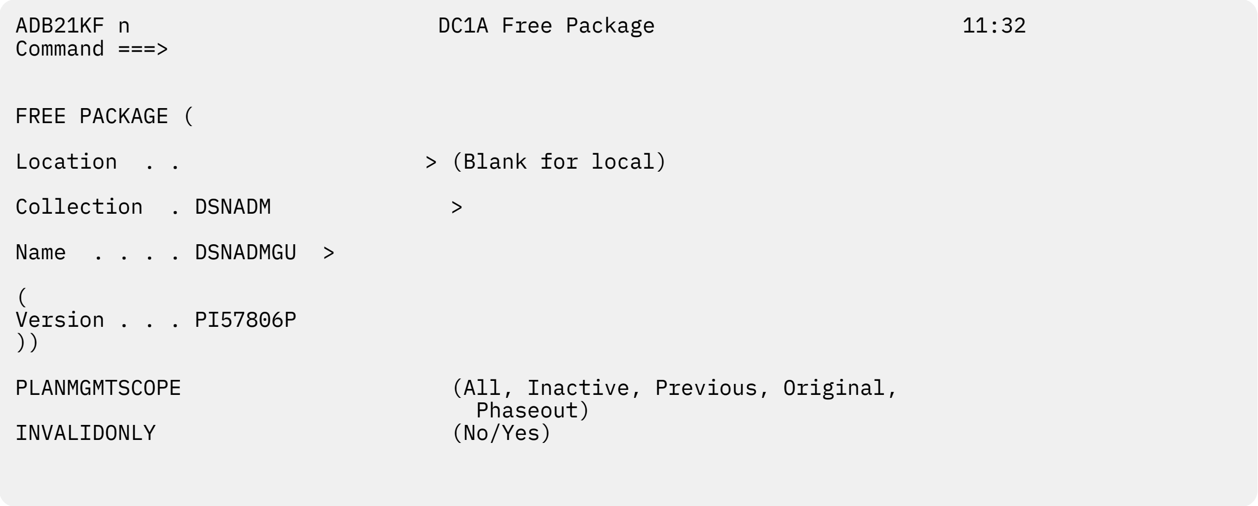
When you free, or delete, a package, the corresponding package information is deleted from the Db2 catalog. You can delete a specific version of a package, all versions of a package, or whole collections of packages.

Procedure

To free a package:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option K (for packages), and press Enter.
3. On the **Packages (ADB21K)** panel, specify the F line command next to the package that you want to free.
4. On the **Free Package (ADB21KF)** panel, specify the options that you want, and press Enter:

   Attention: Use asterisks (*) carefully to avoid unintended consequences. For example, If you specify an asterisk (*) in the **Collection** field, all packages with the specified name and version number are freed. If you specify a collection name and an asterisk (*) in both the **Name** and **Version** fields, all packages in that collection are freed.



```
ADB21KF n                        DC1A Free Package                       11:32
Command ===>


FREE PACKAGE (

Location  . .                   > (Blank for local)

Collection  . DSNADM              >

Name  . . . . DSNADMGU  >

(
Version . . . PI57806P
))

PLANMGMTSCOPE                     (All, Inactive, Previous, Original,
                                    Phaseout)
INVALIDONLY                       (No/Yes)
```

Figure 444. ***Free Package (ADB21KF)*** *panel*

Displaying detailed package information

Db2 Admin Tool can report detailed information for one or more packages, including SQL and EXPLAIN information.

About this task

For each package, this report includes the following two sections:

- Package details, such as the package type and version.
- SQL information and EXPLAIN information, if available. (EXPLAIN information is displayed for only those SQL statements that have EXPLAIN data in the package owner's plan table. EXPLAIN information is also included for queries that are eligible to be offloaded to an accelerator.)

Procedure

To display detailed package information:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option K (for packages), and press Enter.

3. On the **Packages (ADB21K)** panel, specify the DET line command next to the package for which you want to see the details.

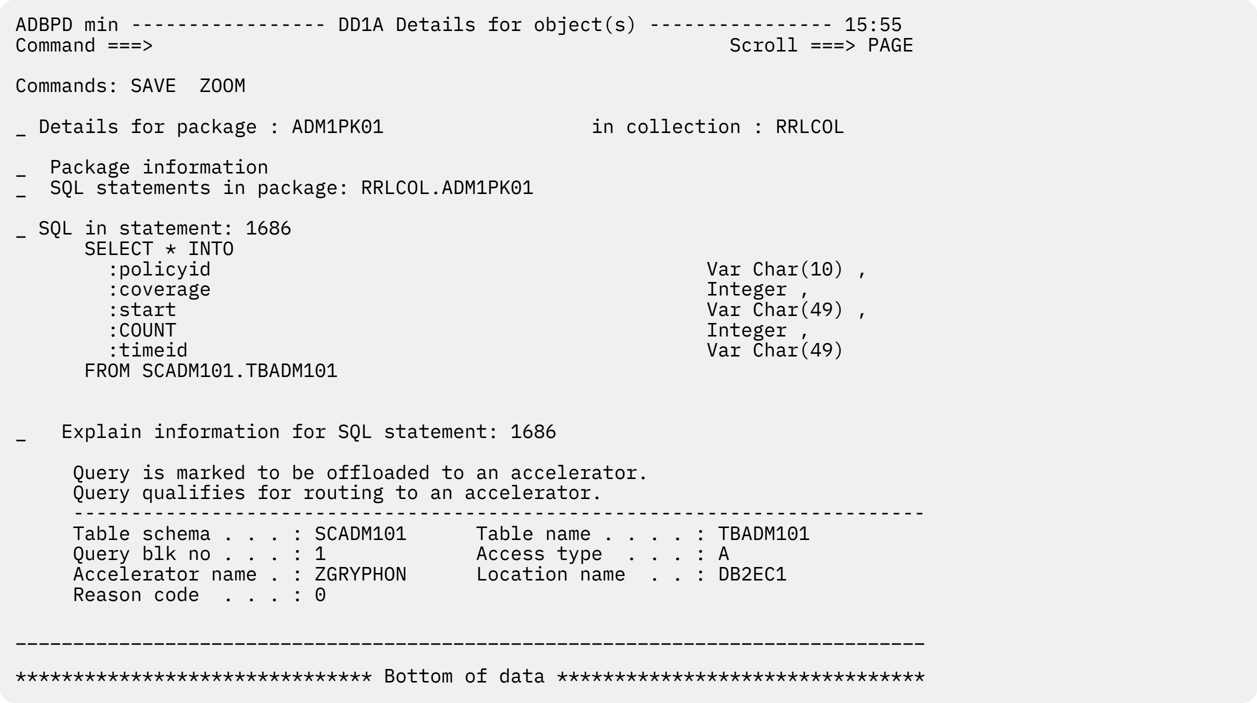
   The package information is displayed on the **Details for object(s) (ADBPD)** panel.

   SQL statements are displayed in lines that are 72 bytes long. If a statement contains host variables, the variable name and data type are displayed on a separate line.

   In the following example of this package information, the SQL information is collapsed.

```
ADBPD                  DD1A Details for object(s)
Command ===>                                                  Scroll ===> PAGE

Commands: SAVE  ZOOM

_ Details for package : SPADJB009012345678901(*1) in collection : SCADJB009(*2)

_  Package information
    Package type . . . . . . . . . . : Native SQL routine package
    Version  . . . . . . . . . . . . : MYVERSION
    Authorization ID of owner  . . . : J148286
    Owner type . . . . . . . . . . . : Auth ID
    Authorization ID of creator  . . : VNDR001
    Created timestamp  . . . . . . . : 2012-08-23-05.38.20.906062
    Latest BIND timestamp  . . . . . : 2012-11-06-16.42.39.648458
    Version under which package bound: V11
    Qualifier for unqualified SQL  . : J148286
    Operative status of package  . . : Package is valid and operative
    Resource and authorization check : At BIND time
    Size of the base section (bytes) : 4272    (in EDM pool during execution)
    Average DML section size (bytes) : 5220    (loaded when needed during exec)
    Package bound with EXPLAIN . . . : Yes
    SQLERROR specified at BIND time  : No - SQLERROR(NOPACKAGE) specified
    BIND or REBIND from remote loc.  : No - (RE)BIND was from a local system
    Remote packages creation method  :
    Source of the package  . . . . . :
    Number of enabled/disabled conn. : 0
    Data concurrency . . . . . . . . : B - not required
      Effect on blocking . . . . . . : Allow blocking for ambiguous cursors
    DEGREE of I/O parallelism  . . . : 1 - parallel I/O inhibited
    Group member that performed BIND :
    Dynamic SQL rules  . . . . . . . : Use definers authid and authorizations
    Re-optimize SQL at execution time: 1 - use exec. time variable values once
    Defer prepare  . . . . . . . . . : Yes - prepare is defered to OPEN time
    Keep prepared dynamic SQL stmts  : No - are destroyed at COMMIT
    Protocol for 3 part names  . . . : D - uses DRDA
    Function resolved at . . . . . . : 2012-11-06-16.42.39.648445
    Optimizer hint identifier  . . . : THIS IS THE OPTHINT FOR JB
    Encode CCSID . . . . . . . . . . : 37
    Write group buffer pool pages  . : Immediate write
    ROUNDING option used on last bind: Round Down
    Concurrent Access  . . . . . . . : W - Wait for release of write lock
    Last date objects used . . . . . : 01/01/0001
    SQL path for resolving UDT,UDF,SP: "J148286","SYSADM","USRT001"

    Precompiler related information:
      Timestamp of precompilation  . : 0001-01-01-00.00.00.000000
      Consistency token in hex . . . : 1941FCD60BBACC4D
      SQL escape character . . . . . : ' (apostrophe)
      Decimal point character  . . . : . (period)
      Host program language  . . . . : Remotely bound, trigger, or SQL package
      Mixed character set  . . . . . : N
      Decimal 31 used  . . . . . . . : Yes
      Katakana . . . . . . . . . . . : No

    Resource allocation information:
      Resources are released . . . . : At plan deallocation time
      Isolation level  . . . . . . . : Read stability

    Temporal special register information:
      Sensitive to SYSTEM_TIME . . . : Yes
      Sensitive to BUSINESS_TIME . . : Yes

    Sensitive to GET_ARCHIVE . . . . : Yes

    Bind options:
      Access path reuse behavior . . : No - does not reuse paths
      Package compat level behavior  : V12R1M502
      Static SQL DESCRIBE requests . : Yes - creates DESCRIBE SQLDA


_  SQL statements in package: SCADJB00901234567890.SPADJB009012345678901234(*3)

_   SQL in statement: 39
_   Explain information for SQL statement: 39
_   SQL in statement: 39
_   SQL in statement: 40
_   SQL in statement: 42
      INSERT INTO SCADJB00.TBADJB00 (ORDER_WAREHOUSE_ID) VALUES ('EEE')


_   Explain information for SQL statement: 42

     The operation is INSERT, UPDATE or DELETE.
     Inner join or no join.
     --------------------------------------------------------------------------
     Table Schema . . . : SCADJB00      Table Name . . . . : TBADJB00
     Query number . . . : 42            Access type  . . . :
     Plan number  . . . : 0             Query block no . . : 1
     Match columns  . . : 0

_   SQL in statement: 39
    CLOSE
     C1

-------------------------------------------------------------------------------
```

The following example shows the displayed package information for a package that contains a query that is marked to be offloaded to an accelerator. Accelerated queries have an access type of A (accesstype = 'A').



```
ADBPD min ---------------- DD1A Details for object(s) --------------- 15:55
Command ===>                                                  Scroll ===> PAGE

Commands: SAVE  ZOOM

_ Details for package : ADM1PK01                 in collection : RRLCOL

_  Package information
_  SQL statements in package: RRLCOL.ADM1PK01

_ SQL in statement: 1686
      SELECT * INTO
        :policyid                                           Var Char(10) ,
        :coverage                                           Integer ,
        :start                                              Var Char(49) ,
        :COUNT                                              Integer ,
        :timeid                                             Var Char(49)
      FROM SCADM101.TBADM101


_   Explain information for SQL statement: 1686

     Query is marked to be offloaded to an accelerator.
     Query qualifies for routing to an accelerator.
     -----------------------------------------------------------------------
     Table schema . . . : SCADM101      Table name . . . . : TBADM101
     Query blk no . . . : 1             Access type  . . . : A
     Accelerator name . : ZGRYPHON      Location name  . . : DB2EC1
     Reason code  . . . : 0

-----------------------------------------------------------------------------
******************************* Bottom of data ********************************
```

Figure 446. ***Details for object(s) (ADBPD)*** *panel*

Viewing SQL statements for a package

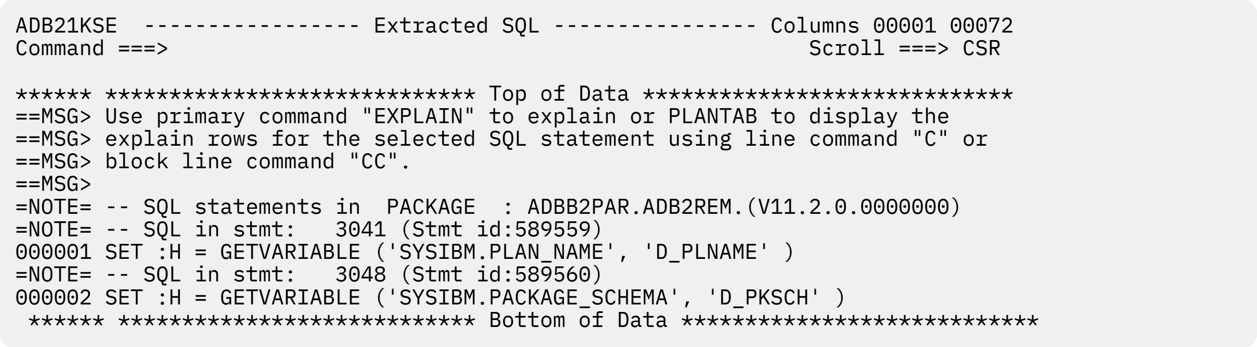
You can use Db2 Admin Tool to view the SQL statements that are included in a package.

Procedure

To view SQL statements for a package:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option K (for packages), and press Enter.
3. On the **Packages (ADB21K)** panel, specify the SQ line command next to the package for which you want to view the SQL statements.

   The **Extracted SQL (ADB21KSE)** panel displays the SQL statements in the package:



```
ADB21KSE  ----------------- Extracted SQL ---------------- Columns 00001 00072
Command ===>                                                  Scroll ===> CSR

****** ****************************** Top of Data *****************************
==MSG> Use primary command "EXPLAIN" to explain or PLANTAB to display the
==MSG> explain rows for the selected SQL statement using line command "C" or
==MSG> block line command "CC".
==MSG>
=NOTE= -- SQL statements in  PACKAGE  : ADBB2PAR.ADB2REM.(V11.2.0.0000000)
=NOTE= -- SQL in stmt:   3041 (Stmt id:589559)
000001 SET :H = GETVARIABLE ('SYSIBM.PLAN_NAME', 'D_PLNAME' )
=NOTE= -- SQL in stmt:   3048 (Stmt id:589560)
000002 SET :H = GETVARIABLE ('SYSIBM.PACKAGE_SCHEMA', 'D_PKSCH' )
 ****** **************************** Bottom of Data ****************************
```

Figure 447. ***Extracted SQL (ADB21KSE)*** *panel*

4. Optional: To get EXPLAIN information for one or more of the listed SQL statements, issue the C line command or the CC block line command and specify one of the following primary commands:

EXPLAIN
Explains the selected SQL statement.

PLANTAB
Displays EXPLAIN rows for the selected SQL statement.

The **EXPLAIN (ADB2E)** panel is displayed.

Viewing SQL statements for a collection

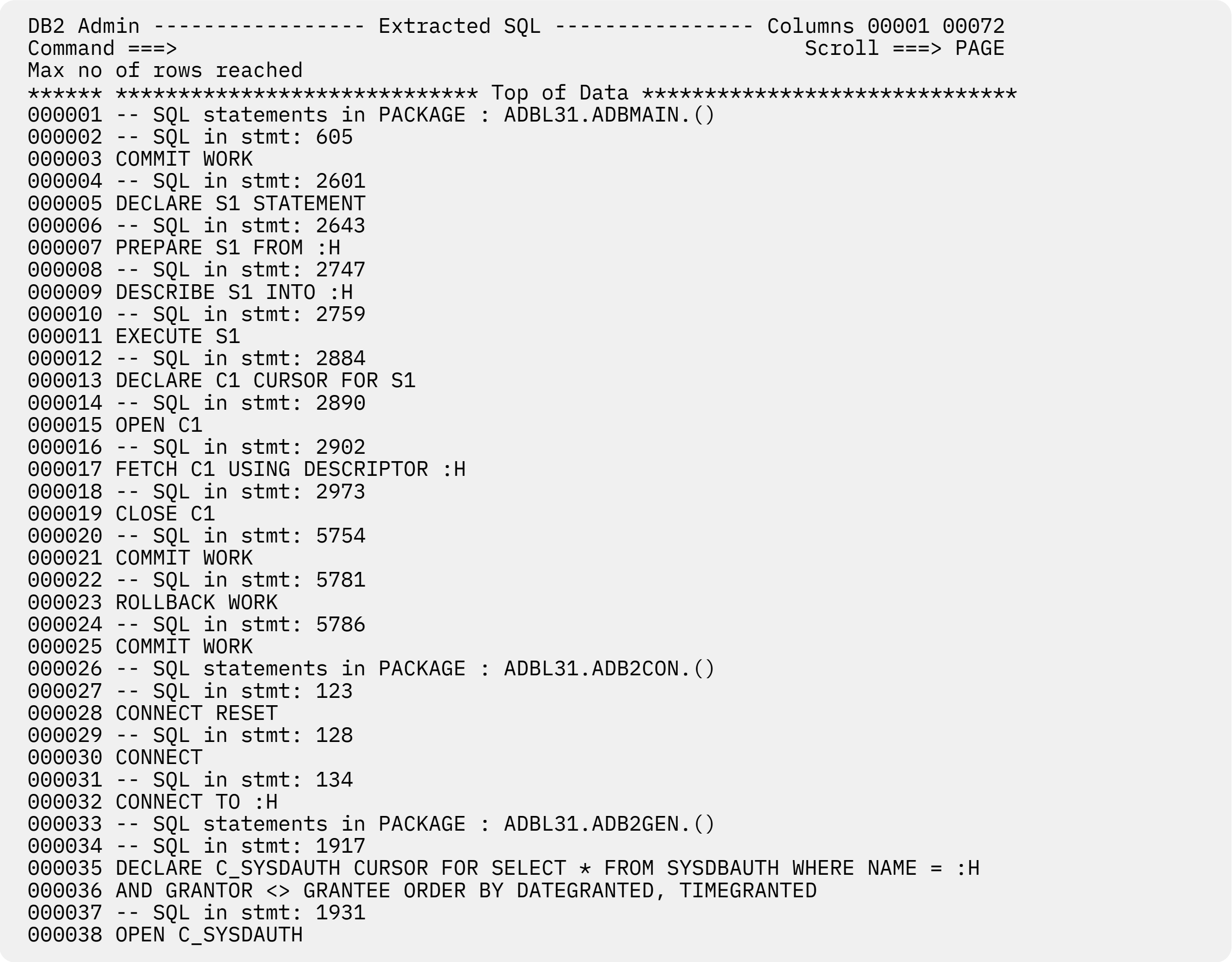
You can use Db2 Admin Tool to view the SQL statements that are included in a collection of packages.

Procedure

To view SQL statements for a collection:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option L (for collections), and press Enter.
3. On the **Collections (ADB21L)** panel, specify the SQ line command next to the collection for which you want to view the SQL statements, and press Enter:

   The **Extracted SQL (ADB21KSE)** panel displays the SQL statements for each package in the collection:



```
DB2 Admin ----------------- Extracted SQL ---------------- Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
Max no of rows reached
****** ****************************** Top of Data ******************************
000001 -- SQL statements in PACKAGE : ADBL31.ADBMAIN.()
000002 -- SQL in stmt: 605
000003 COMMIT WORK
000004 -- SQL in stmt: 2601
000005 DECLARE S1 STATEMENT
000006 -- SQL in stmt: 2643
000007 PREPARE S1 FROM :H
000008 -- SQL in stmt: 2747
000009 DESCRIBE S1 INTO :H
000010 -- SQL in stmt: 2759
000011 EXECUTE S1
000012 -- SQL in stmt: 2884
000013 DECLARE C1 CURSOR FOR S1
000014 -- SQL in stmt: 2890
000015 OPEN C1
000016 -- SQL in stmt: 2902
000017 FETCH C1 USING DESCRIPTOR :H
000018 -- SQL in stmt: 2973
000019 CLOSE C1
000020 -- SQL in stmt: 5754
000021 COMMIT WORK
000022 -- SQL in stmt: 5781
000023 ROLLBACK WORK
000024 -- SQL in stmt: 5786
000025 COMMIT WORK
000026 -- SQL statements in PACKAGE : ADBL31.ADB2CON.()
000027 -- SQL in stmt: 123
000028 CONNECT RESET
000029 -- SQL in stmt: 128
000030 CONNECT
000031 -- SQL in stmt: 134
000032 CONNECT TO :H
000033 -- SQL statements in PACKAGE : ADBL31.ADB2GEN.()
000034 -- SQL in stmt: 1917
000035 DECLARE C_SYSDAUTH CURSOR FOR SELECT * FROM SYSDBAUTH WHERE NAME = :H
000036 AND GRANTOR <> GRANTEE ORDER BY DATEGRANTED, TIMEGRANTED
000037 -- SQL in stmt: 1931
000038 OPEN C_SYSDAUTH
```

Figure 448. ***Extracted SQL (ADB21KSE)*** *panel*

Viewing information about package copies

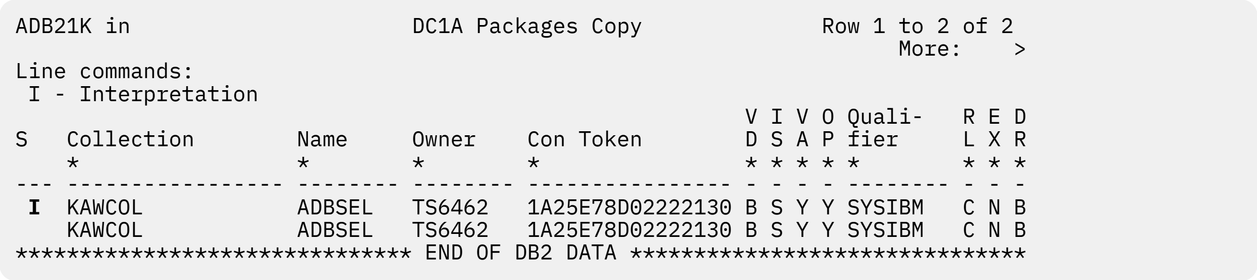
When you rebind packages with the PLANMGMT bind option set to EXTENDED or BASIC, Db2 saves copies of the package.

Procedure

To view information about package copies:

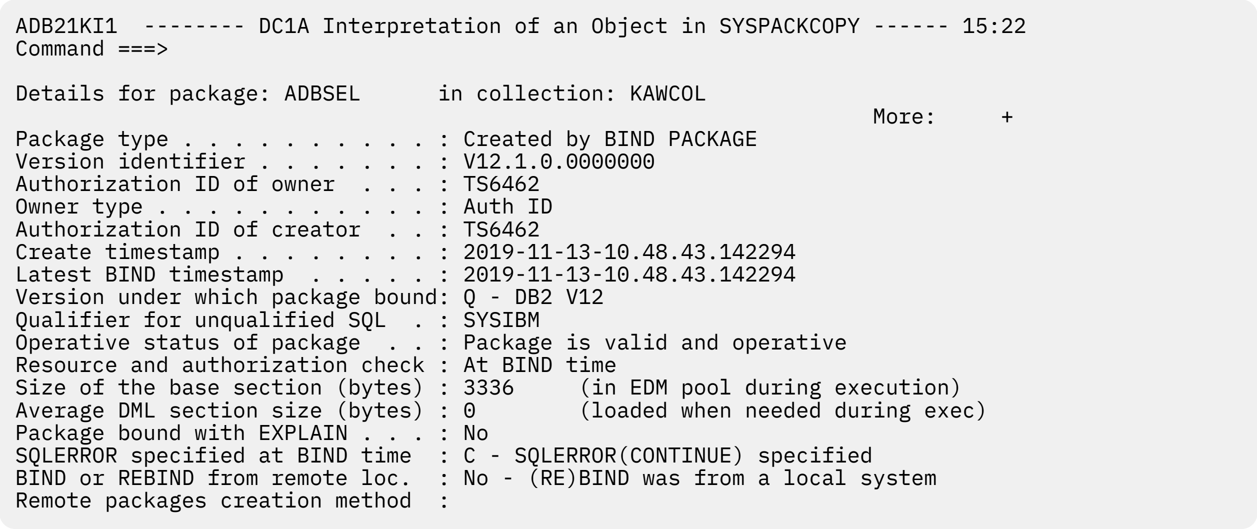
1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option K, and press Enter.
3. On the **Packages (ADB21K)** panel, specify the KC line command next to the package for which you want to view copies.

   The KC line command is valid only if copies of the package exist.
4. On the **Packages Copy (ADB21K)** panel, specify the I line command next to the package copy for which you want to view the details:



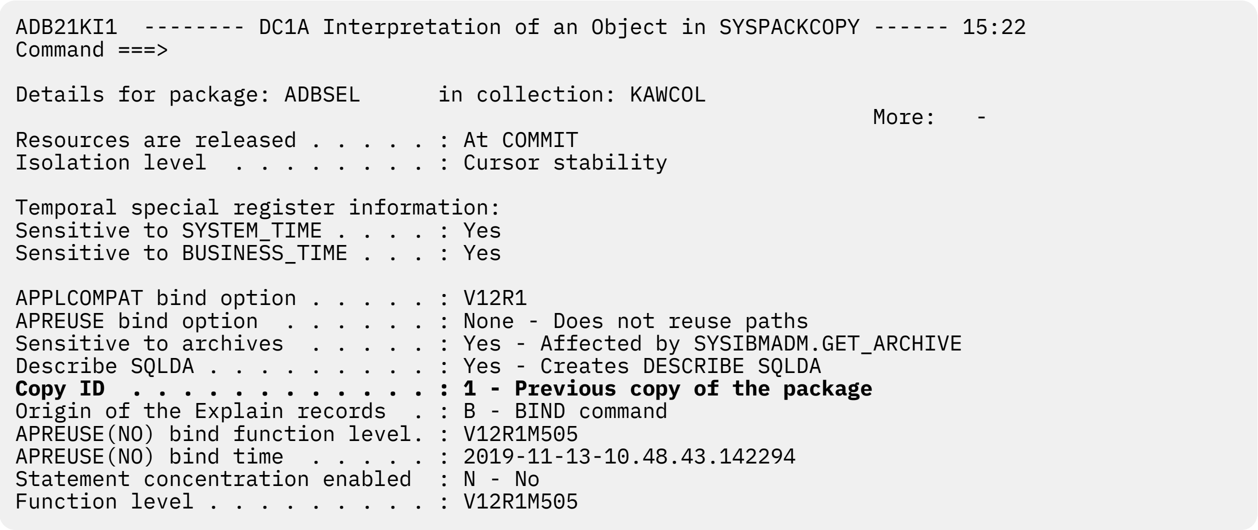
```
ADB21K in                      DC1A Packages Copy              Row 1 to 2 of 2
                                                                     More:    >
Line commands:
 I - Interpretation
                                                         V I V O Quali-   R E D
S   Collection        Name     Owner    Con Token        D S A P fier     L X R
    *                 *        *        *                * * * * *        * * *
--- ----------------- -------- -------- ---------------- - - - - -------- - - -
 I  KAWCOL            ADBSEL   TS6462   1A25E78D02222130 B S Y Y SYSIBM   C N B
    KAWCOL            ADBSEL   TS6462   1A25E78D02222130 B S Y Y SYSIBM   C N B
******************************* END OF DB2 DATA *******************************
```

   The **Interpretation of an Object (ADB21KI1)** panel displays detailed information about the package copy from the SYSPACKCOPY catalog table:



```
ADB21KI1  -------- DC1A Interpretation of an Object in SYSPACKCOPY ------ 15:22
Command ===>

Details for package: ADBSEL      in collection: KAWCOL
                                                                   More:     +
Package type . . . . . . . . . . : Created by BIND PACKAGE
Version identifier . . . . . . . : V12.1.0.0000000
Authorization ID of owner  . . . : TS6462
Owner type . . . . . . . . . . . : Auth ID
Authorization ID of creator  . . : TS6462
Create timestamp . . . . . . . . : 2019-11-13-10.48.43.142294
Latest BIND timestamp  . . . . . : 2019-11-13-10.48.43.142294
Version under which package bound: Q - DB2 V12
Qualifier for unqualified SQL  . : SYSIBM
Operative status of package  . . : Package is valid and operative
Resource and authorization check : At BIND time
Size of the base section (bytes) : 3336     (in EDM pool during execution)
Average DML section size (bytes) : 0        (loaded when needed during exec)
Package bound with EXPLAIN . . . : No
SQLERROR specified at BIND time  : C - SQLERROR(CONTINUE) specified
BIND or REBIND from remote loc.  : No - (RE)BIND was from a local system
Remote packages creation method  :
```

5. To determine which copy this panel describes, scroll to the **Copy ID** field:



```
ADB21KI1  -------- DC1A Interpretation of an Object in SYSPACKCOPY ------ 15:22
Command ===>

Details for package: ADBSEL      in collection: KAWCOL
                                                                    More:   -
Resources are released . . . . . : At COMMIT
Isolation level  . . . . . . . . : Cursor stability

Temporal special register information:
Sensitive to SYSTEM_TIME . . . . : Yes
Sensitive to BUSINESS_TIME . . . : Yes

APPLCOMPAT bind option . . . . . : V12R1
APREUSE bind option  . . . . . . : None - Does not reuse paths
Sensitive to archives  . . . . . : Yes - Affected by SYSIBMADM.GET_ARCHIVE
Describe SQLDA . . . . . . . . . : Yes - Creates DESCRIBE SQLDA
Copy ID  . . . . . . . . . . . . : 1 - Previous copy of the package
Origin of the Explain records  . : B - BIND command
APREUSE(NO) bind function level. : V12R1M505
APREUSE(NO) bind time  . . . . . : 2019-11-13-10.48.43.142294
Statement concentration enabled  : N - No
Function level . . . . . . . . . : V12R1M505
```

Related information

Package copies for plan management (Db2 12 for z/OS)

SYSPACKCOPY catalog table (Db2 12 for z/OS)

Deleting obsolete packages

Deleting obsolete packages helps clean up your package collections and the Db2 catalog.

Before you begin

Requirement: You can clean up packages and catalog objects only for COBOL or PL/I programs. Other languages are not supported.

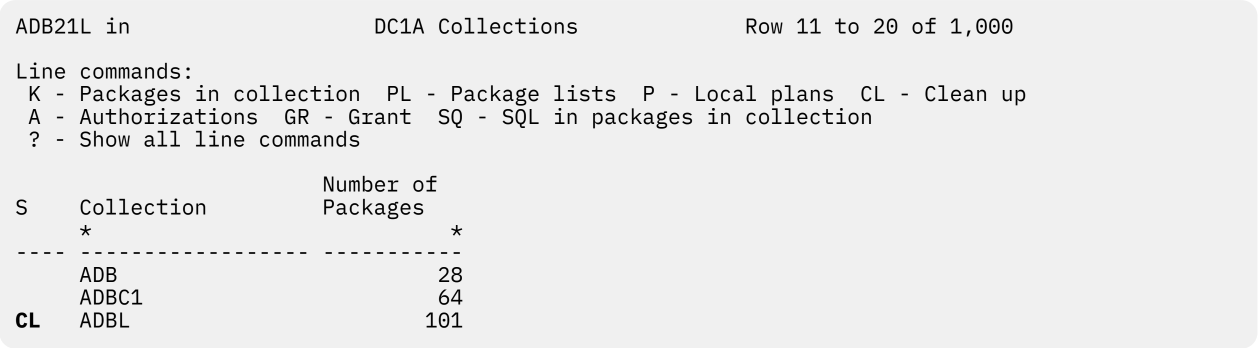
About this task

Db2 Admin Tool considers packages to be obsolete if they are not referenced by load libraries. When you delete, or free, these packages, the corresponding package entries are deleted from the Db2 catalog.

Procedure

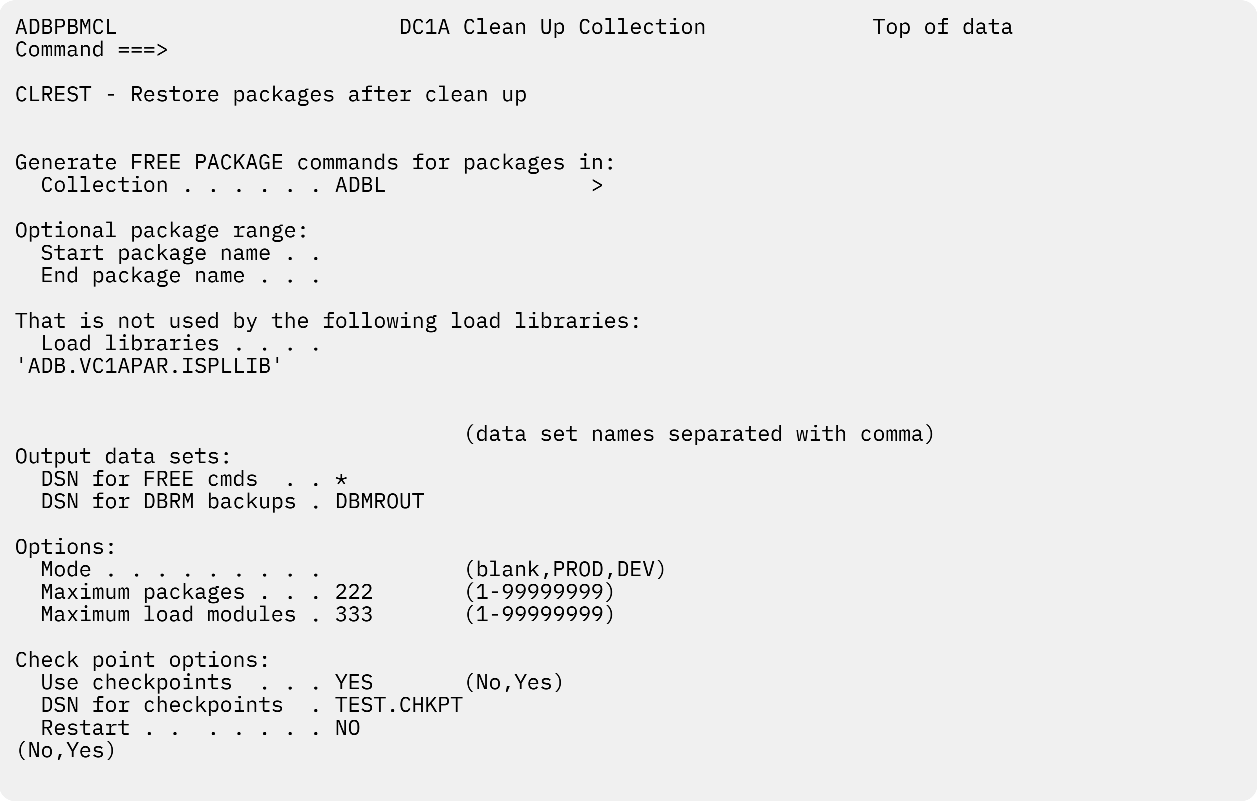
To delete an obsolete package:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option L (for collections), and press Enter.
3. On the **Collections (ADB21L)** panel, specify the CL line command next to the collection that you want to clean up, and press Enter:



```
ADB21L in                   DC1A Collections            Row 11 to 20 of 1,000

Line commands:
 K - Packages in collection  PL - Package lists  P - Local plans  CL - Clean up
 A - Authorizations  GR - Grant  SQ - SQL in packages in collection
 ? - Show all line commands

                         Number of
S    Collection          Packages
     *                          *
---- ------------------ ----------
     ADB                        28
     ADBC1                      64
CL   ADBL                      101
```

4. On the **Clean up Collection (ADBPBMCL)** panel, specify the load libraries that you want checked for references to the packages in the collection. Also specify the output data set names and any other options that you want, and press Enter:



```
ADBPBMCL                      DC1A Clean Up Collection             Top of data
Command ===>

CLREST - Restore packages after clean up


Generate FREE PACKAGE commands for packages in:
  Collection . . . . . . ADBL                >

Optional package range:
  Start package name . .
  End package name . . .

That is not used by the following load libraries:
  Load libraries . . . .
'ADB.VC1APAR.ISPLLIB'


                                   (data set names separated with comma)
Output data sets:
  DSN for FREE cmds  . . *
  DSN for DBRM backups . DBMROUT

Options:
  Mode . . . . . . . . .           (blank,PROD,DEV)
  Maximum packages . . . 222       (1-99999999)
  Maximum load modules . 333       (1-99999999)

Check point options:
  Use checkpoints  . . . YES       (No,Yes)
  DSN for checkpoints  . TEST.CHKPT
  Restart . .  . . . . . NO
(No,Yes)
```

A generated job is displayed. This job checks the specified load libraries for references to the packages in the specified collection. For any packages that are not referenced, this job generates FREE commands and writes them to the specified output data set.

5. Submit the generated job and ensure that it ran successfully.
6. Run any generated FREE commands.

Restoring packages

You can restore one or more packages that were freed by the collection clean up function.

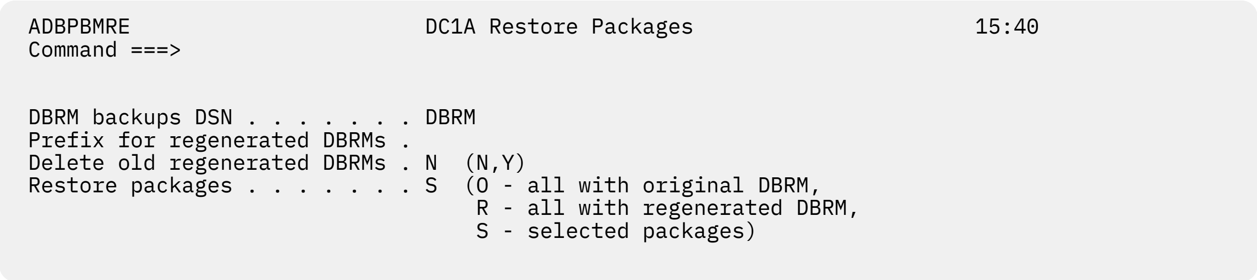
About this task

The *collection clean up function* runs when you specify the CL line command on the **Collections (ADB21L)** panel, as described in "Deleting obsolete packages" on page 925.

Procedure

To restore packages:

1. Specify the CLREST command.
2. On the **Restore Packages (ADBPBMRE)** panel, specify the requested information to restore the packages, and press Enter:

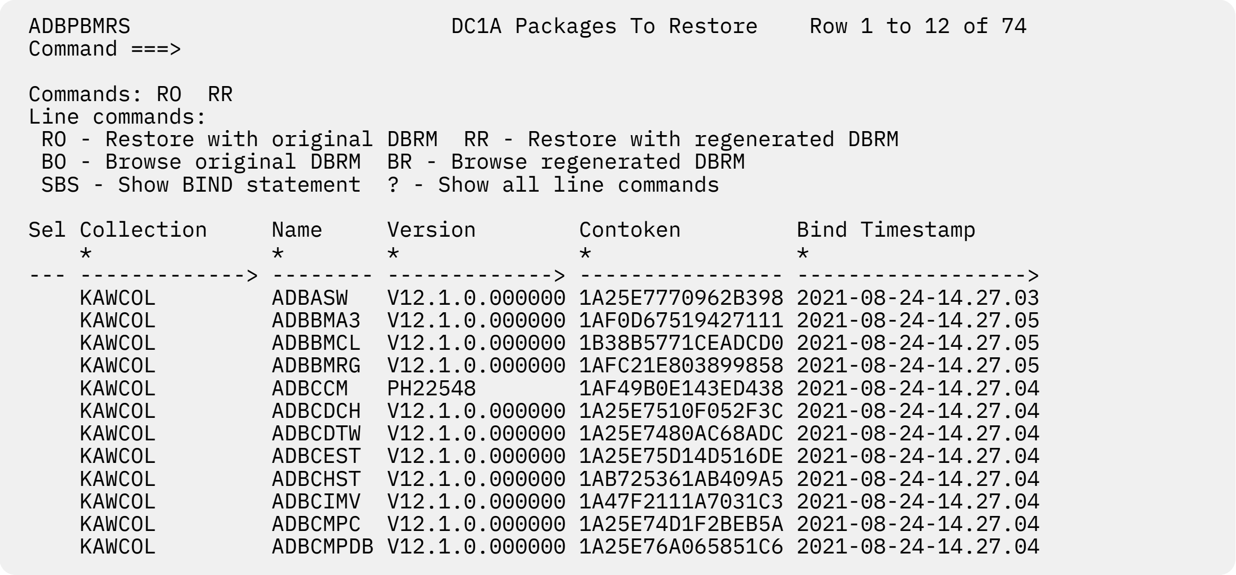


```
ADBPBMRE                       DC1A Restore Packages                      15:40
Command ===>


DBRM backups DSN . . . . . . . DBRM
Prefix for regenerated DBRMs .
Delete old regenerated DBRMs . N  (N,Y)
Restore packages . . . . . . . S  (O - all with original DBRM,
                                   R - all with regenerated DBRM,
                                   S - selected packages)
```

Tip: For more information about the fields on this panel, press PF1 to open the panel help.

After you press Enter, one of the following screens is displayed:

- If you specified O or R in the **Restore packages** field, the generated JCL to restore the packages is displayed. This JCL contains the appropriate BIND statements for the requested packages.
- If you specified S in the **Restore packages** field, the **Packages To Restore (ADBPBMRS)** panel is displayed:



```
ADBPBMRS                         DC1A Packages To Restore    Row 1 to 12 of 74
Command ===>

Commands: RO  RR
Line commands:
 RO - Restore with original DBRM  RR - Restore with regenerated DBRM
 BO - Browse original DBRM  BR - Browse regenerated DBRM
 SBS - Show BIND statement  ? - Show all line commands

Sel Collection     Name     Version        Contoken         Bind Timestamp
    *              *        *              *                *
--- -------------> -------- -------------> ---------------- ------------------>
    KAWCOL         ADBASW   V12.1.0.000000 1A25E7770962B398 2021-08-24-14.27.03
    KAWCOL         ADBBMA3  V12.1.0.000000 1AF0D67519427111 2021-08-24-14.27.05
    KAWCOL         ADBBMCL  V12.1.0.000000 1B38B5771CEADCD0 2021-08-24-14.27.05
    KAWCOL         ADBBMRG  V12.1.0.000000 1AFC21E803899858 2021-08-24-14.27.05
    KAWCOL         ADBCCM   PH22548        1AF49B0E143ED438 2021-08-24-14.27.04
    KAWCOL         ADBCDCH  V12.1.0.000000 1A25E7510F052F3C 2021-08-24-14.27.04
    KAWCOL         ADBCDTW  V12.1.0.000000 1A25E7480AC68ADC 2021-08-24-14.27.04
    KAWCOL         ADBCEST  V12.1.0.000000 1A25E75D14D516DE 2021-08-24-14.27.04
    KAWCOL         ADBCHST  V12.1.0.000000 1AB725361AB409A5 2021-08-24-14.27.04
    KAWCOL         ADBCIMV  V12.1.0.000000 1A47F2111A7031C3 2021-08-24-14.27.04
    KAWCOL         ADBCMPC  V12.1.0.000000 1A25E74D1F2BEB5A 2021-08-24-14.27.04
    KAWCOL         ADBCMPDB V12.1.0.000000 1A25E76A065851C6 2021-08-24-14.27.04
```

3. If the **Packages To Restore (ADBPBMRS)** panel is displayed, use the following commands to select the packages that you want to bind:

   RO

   Restore packages by using the original DBRM.

   If you specify the RO line command, the appropriate BIND statement is executed for the selected package. (Depending on your settings, an execution prompt might be displayed first.)

   If you specify the RO command, JCL is generated to restore all packages that are currently listed on the panel. To limit the scope of this command, you can first filter the list of packages displayed on the panel.

   RR

   Restore packages by using the regenerated DBRM.

   If you specify the RR line command, JCL is generated to restore the package. This JCL contains the appropriate BIND statement for the selected package.

   If you specify the RR command, JCL is generated to restore all packages that are currently listed on the panel. To limit the scope of this command, you can first filter the list of packages displayed on the panel.
4. Submit any generated JCL to bind the packages.

Regenerating DBRMs

Db2 Admin Tool can regenerate any missing DBRMs for you based on the information in the Db2 catalog table SYSIBM.SYSPACKSTMT.

Before you begin

Requirement: You can regenerate only those DBRMs that were created by the Db2 precompiler or coprocessor for application programs that are written in the following languages:

- ASSEMBLER

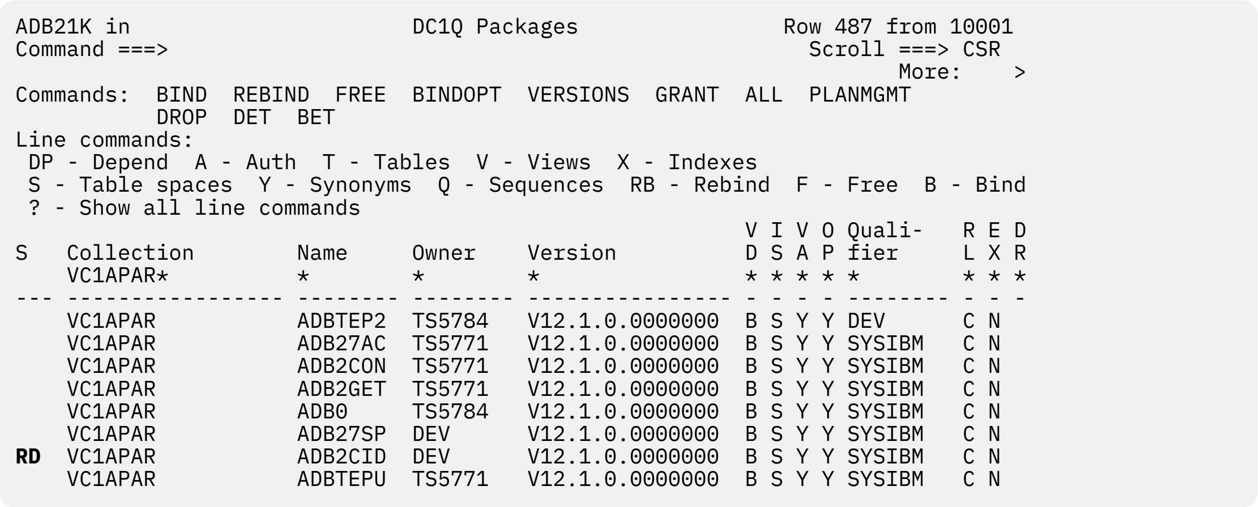
- COBOL
- COBOL_II
- IBM_COBOL
- FORTRAN
- PL/I
- C
- C++
- SQLJ

Other languages are not supported.

Procedure

To regenerate DBRMs:

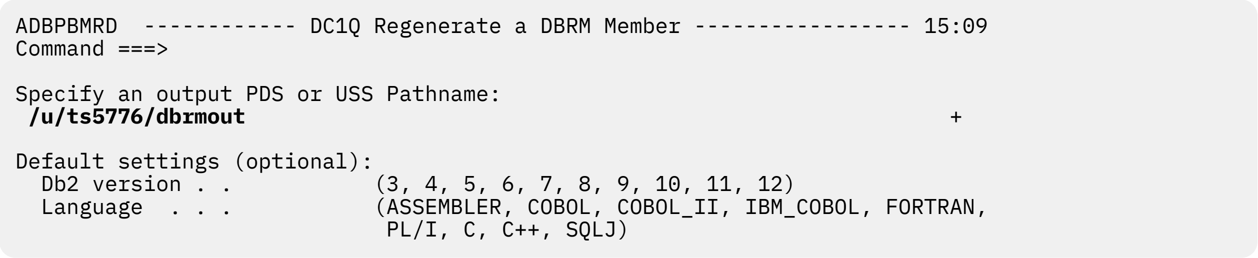
1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option K (for packages), and press Enter.
3. On the **Packages (ADB21K)** panel, specify the RD line command next to the package for which you want to regenerate the associated DBRM, and press Enter:



```
ADB21K in                      DC1Q Packages                Row 487 from 10001
Command ===>                                                  Scroll ===> CSR
                                                                     More:    >
Commands:  BIND  REBIND  FREE  BINDOPT  VERSIONS  GRANT  ALL  PLANMGMT
           DROP  DET  BET
Line commands:
 DP - Depend  A - Auth  T - Tables  V - Views  X - Indexes
 S - Table spaces  Y - Synonyms  Q - Sequences  RB - Rebind  F - Free  B - Bind
 ? - Show all line commands
                                                         V I V O Quali-   R E D
S   Collection        Name     Owner    Version          D S A P fier     L X R
    VC1APAR*          *        *        *                * * * * *        * * *
--- ----------------- -------- -------- ---------------- - - - - -------- - - -
    VC1APAR           ADBTEP2  TS5784   V12.1.0.0000000  B S Y Y DEV      C N
    VC1APAR           ADB27AC  TS5771   V12.1.0.0000000  B S Y Y SYSIBM   C N
    VC1APAR           ADB2CON  TS5771   V12.1.0.0000000  B S Y Y SYSIBM   C N
    VC1APAR           ADB2GET  TS5771   V12.1.0.0000000  B S Y Y SYSIBM   C N
    VC1APAR           ADB0     TS5784   V12.1.0.0000000  B S Y Y SYSIBM   C N
    VC1APAR           ADB27SP  DEV      V12.1.0.0000000  B S Y Y SYSIBM   C N
RD  VC1APAR           ADB2CID  DEV      V12.1.0.0000000  B S Y Y SYSIBM   C N
    VC1APAR           ADBTEPU  TS5771   V12.1.0.0000000  B S Y Y SYSIBM   C N
```

4. On the **Regenerate a DBRM Member (ADBPBMRD)** panel, specify where the regenerated DBRM member is to be stored. You can specify either a partitioned data set (PDS) or z/OS UNIX System Services (USS) path name. The path name must already exist.

   Example of specifying a USS path name:



```
ADBPBMRD  ------------ DC1Q Regenerate a DBRM Member ----------------- 15:09
Command ===>

Specify an output PDS or USS Pathname:
 /u/ts5776/dbrmout                                                      +

Default settings (optional):
  Db2 version . .           (3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
  Language  . . .           (ASSEMBLER, COBOL, COBOL_II, IBM_COBOL, FORTRAN,
                             PL/I, C, C++, SQLJ)
```

   If you specify a USS path name, the package name is appended to the path name.

   Example of specifying a PDS:

```
ADBPBMRD  ------------ DC1Q Regenerate a DBRM Member ---------------- 15:09
Command ===>

Specify an output PDS or USS Pathname:
 'TS5776.DBRMOUT'                                                    +

Default settings (optional):
  Db2 version . .           (3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
  Language  . . .           (ASSEMBLER, COBOL, COBOL_II, IBM_COBOL, FORTRAN,
                             PL/I, C, C++, SQLJ)
```

If you specify a PDS and the data set is not cataloged, you are prompted whether you want Db2 Admin Tool to create the data set for you.

Db2 Admin Tool generates a batch job to regenerate the DBRM.

5. Submit the generated job and ensure that it ran successfully.

IBM Db2 Analytics Accelerator for z/OS

IBM Db2 Analytics Accelerator for z/OS (IDAA)is an optional workload-optimized appliance add-on that is integrated with Db2 for z/OS. IBM Db2 Analytics Accelerator for z/OS maximizes performance for long-running complex queries while reducing processor usage.

IBM Db2 Analytics Accelerator for z/OS is a combined hardware and software solution offered by Netezza® that significantly reduces response times for Db2 for z/OS SQL queries. The components work together to support a variety of data analysis and business reporting tasks. The Db2 Analytics Accelerator maximizes performance for long-running complex queries in magnitudes while reducing CPU utilization on System z®. It also offers high performance storage saver capability that enables storing historic data in Db2 tables or partitions solely on the accelerator.

The Db2 Analytics Accelerator acts as an additional access path for Db2 for z/OS. Whenever queries are eligible for being processed by the Db2 Analytics Accelerator , users will immediately benefit from shortened response times without any further actions.

You can use Db2 Admin Tool to customize parameters for use with IBM Db2 Analytics Accelerator for z/OS.

In addition to basic accelerator functions, you can also perform the following accelerator tasks:

- You can create both real and virtual accelerators. Virtual accelerators do not require accelerator hardware, so you can use them to determine whether queries can be accelerated, check queries for errors, and estimate query response times.
- You can test the connection between Db2 and the Db2 Analytics Accelerator .
- You can display or cancel active accelerator tasks.
- You can display, add, delete, or activate trace profiles.
- You can save accelerator trace data.

Prerequisites: Before you can use Db2 Analytics Accelerator (IDAA) with Db2 Admin Tool, ensure that the following requirements are met:

- Db2 Analytics Accelerator is installed and configured for an existing Db2 subsystem. See Enabling an existing Db2 subsystem for IBM Db2 Analytics Accelerator for z/OS (IBM DB2 Analytics Accelerator for z/OS 7.5). This process creates and configures all the required objects on Db2 (such as tables and stored procedures) that are required for processing IDAA-related scenarios in Db2 Admin Tool.
- IDAA is enabled in Db2 Admin Tool.

  This enablement is specified during customization with TCz. Specifically, on the **Product Parameters (CCQPPRD)** panel, the **IDAA** field must be set to YES:

```
*IDAA . . . . . . . . . . . . . . . . . . . . YES (YES, NO)
 Owner  . . . . . . . . . . . . . . . . . .                               >
 GRANT EXECUTE ON PROCEDURE TO  . . . . . .                               >
 GRANT SELECT ON TABLE TO . . . . . . . . .                               >
```

Optionally, the **Owner** and two **GRANT** fields can also be set to control authorization for the IDAA objects. For details, see the field help (PF1).

If IDAA is not already enabled, recustomize Db2 Admin Tool and set **IDAA** to YES. Then, regenerate the TCz jobs and resubmit them in order. For detailed instructions, see "Roadmap: Recustomizing Db2 Admin Tool" on page 96.

If **IDAA** is set to YES during customization, but Db2 Analytics Accelerator is not installed, you can ignore certain errors from the TCz job ADBBIND, as described in the job comments. However, IDAA-related scenarios in Db2 Admin Tool will not be available to users until IDAA is installed, configured, and the ADBBIND job is resubmitted.

Adding accelerators

You can add a real accelerator or you can add a virtual accelerator for testing purposes.

About this task

Virtual accelerators use the EXPLAIN function offered by Db2 for z/OS. Virtual accelerators cannot process regular queries and cannot return query results. However, because virtual accelerators do not require accelerator hardware, you can use them to determine whether queries can be accelerated, check queries for errors, and estimate query response times. Virtual accelerators must be started with the ACCESS(EXPLAINONLY) statement, and can accept only queries that contain the EXPLAIN statement.

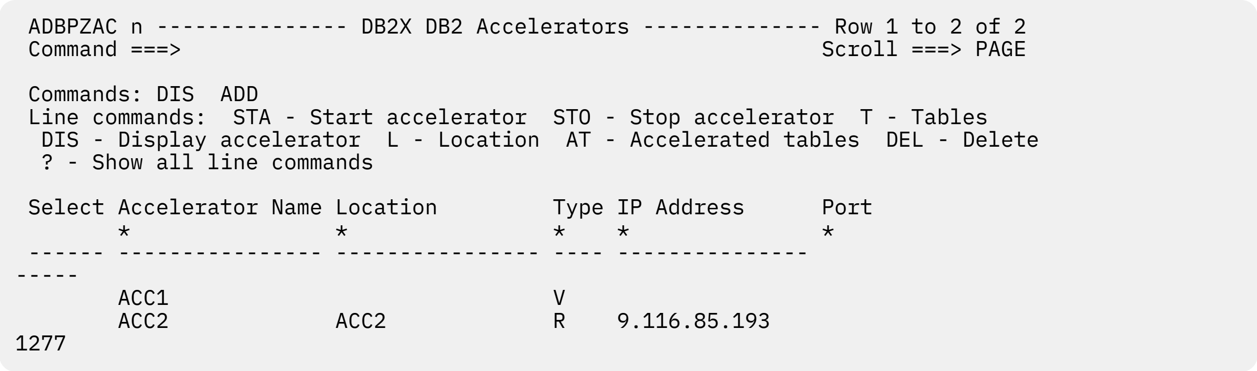
Requirement: After you add a real or virtual accelerator, you must issue the -START ACCEL command to make the accelerator functional.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed:



```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ ---------------- ---------------- ---- ---------------
-----
        ACC1                              V
        ACC2             ACC2             R    9.116.85.193
1277
```

3. Specify the ADD command, and press Enter.

   The **Add Accelerator (ADBPZACA)** panel is displayed:

```
ADBPZACA  --------------------- DB2X Add Accelerator ----------------
Command ===>

Enter accelerator details:

Accelerator name  . .
IP address  . . . . .                     > (IPv4 or IPv6 address)
Port  . . . . . . . .                       (numeric)
Pin . . . . . . . . .                       (numeric)
Location  . . . . . .                       (? to lookup)

Start accelerator . .                       (Yes/No, Default - Yes)

Press ENTER to Add accelerator, or PF3 to cancel Add.
```

4. Determine whether you want to create a real accelerator or a virtual accelerator, and complete the steps.

   - To create a real accelerator, specify values for all of the parameters, and press Enter.
   - To create a virtual accelerator, specify a value only for the **Accelerator name** field, and press Enter.

   If the accelerator was added successfully, the accelerator information is added to the SYSACCEL.SYSACCELERATORS table and the following message is displayed:

   ```
   Insert stmt executed
   ```

Defining an accelerator group

Define an accelerator group when you want an accelerator-only table to be defined on more than one accelerator.

About this task

When you define an accelerator group, you define a location alias that you can use to represent multiple accelerators. That location alias is defined in the Db2 catalog table SYSIBM.LOCATIONS.

Procedure

To define an accelerator group:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option DU, and press Enter.
3. On the **Display/Update CDB** panel, specify option L, and press Enter.
4. On the **Display/Update LOCATIONS** panel, specify the I line command next to any location.
5. On the **Insert LOCATION (ADB2Z5LU)** panel, specify the following information and press Enter:

   Location
   The accelerator group name. This name must not exceed eight characters and must consist of the characters A-Z and 0-9.

   Link name
   DSNACCELERATORALIAS

   DBALIAS
   The names of the accelerators that you want to include in this group.

```
ADB2Z5LU  ------------------- DD1A Update LOCATION ---------------------- 14:33
Command ===>

                                                         DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
Enter/verify:
  Location . . . . . . ACCALIAS          > ( Location of the Remote Server    )
  Link name  . . . . . DSNACCELERATORALIAS  > (LU name, IP link name for remote
                                               system or DSNACCELERATORALIAS  )
  Port . . . . . . . .                                     > (TCP/IP Port     )
  TP name  . . . . . .                                     > (SNA Program name)
  DBALIAS  . . . . . . IDAAS08 IDAAZ12                     > (Remote address  )
  TRUSTED  . . . . . .            (Yes/No)
  SECURE . . . . . . .            (Yes/No)

Press ENTER to Update LOCATION, or PF3 to cancel Update.
```

6. If the **Statement Execution Prompt (ADB2PSTM)** panel is displayed, specify 1, and press Enter to execute the INSERT statement.
7. Verify that the new location alias is displayed on the **Display/Update LOCATIONS** panel:

```
ADB2Z5L n ------------ DD1A Display/Update LOCATIONS -------- Row 147 from 149

                                                         DB2 System: DD1A
Line commands:
 D - Delete  I - Insert  U - Update  DIS - Display location  S - Select
 ALIAS - Aliases for location  LU - LU name  IP - IP name
 ILU - Insert LU  IIP - Insert IP name

                        Link
Select Location         Name     Port       TP Name  DBALIAS   TRUSTED SECURE
       *                DSNA*    *          *        *         *       *
------ ---------------> -------> ---------> -------> --------> ------- ------
       ACCALIAS         DSNACCEL                     IDAAS08 I
       TESTALS          DSNACCEL                     IDAAS08 I
       ACTESTAS         DSNACCEL                     IDAAS08
****************************** END OF DB2 DATA ******************************
```

What to do next

When you create or alter an accelerator-only table, you can now specify this accelerator group (location alias). The accelerator-only table is then defined in all accelerators that are associated with the group.

Related information

LOCATIONS catalog table (Db2 12 for z/OS documentation)

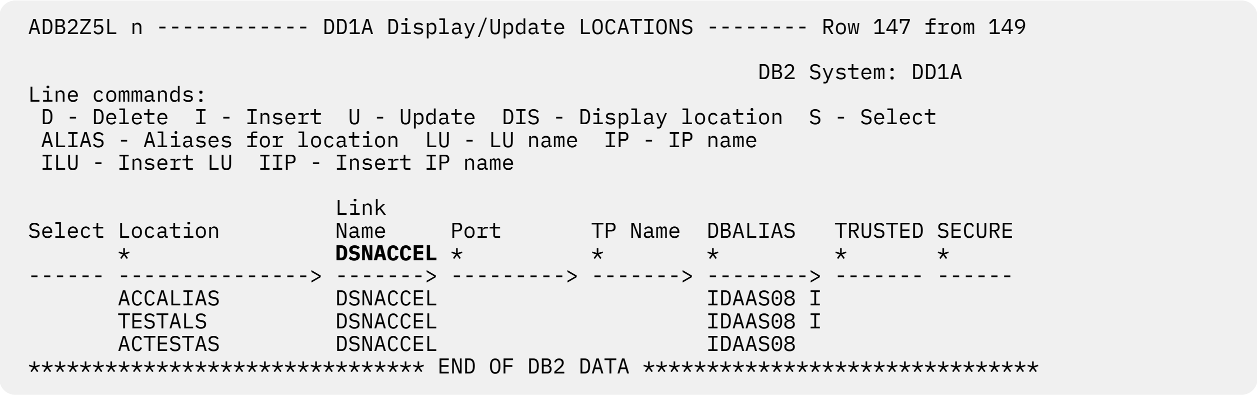
Modifying an accelerator group

You can change an accelerator group to remove or add an accelerator.

Procedure

To modify an accelerator group:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option DU, and press Enter.
3. On the **Display/Update CDB** panel, specify option L, and press Enter.
4. On the **Display/Update LOCATIONS** panel, find the accelerator group that you want to modify. You can filter this panel to display only accelerator groups by specifying DSNACCEL in the search arguments field under the **Link Name** column:



```
ADB2Z5L n ------------ DD1A Display/Update LOCATIONS -------- Row 147 from 149

                                                         DB2 System: DD1A
Line commands:
 D - Delete  I - Insert  U - Update  DIS - Display location  S - Select
 ALIAS - Aliases for location  LU - LU name  IP - IP name
 ILU - Insert LU  IIP - Insert IP name

                        Link
Select Location         Name     Port       TP Name  DBALIAS   TRUSTED SECURE
       *                DSNACCEL *          *        *         *       *
------ ---------------> -------> ---------> -------> --------> ------- ------
       ACCALIAS         DSNACCEL                     IDAAS08 I
       TESTALS          DSNACCEL                     IDAAS08 I
       ACTESTAS         DSNACCEL                     IDAAS08
******************************* END OF DB2 DATA ********************************
```

5. Specify the U line command next to the group that you want to update.
6. On the **Update LOCATION (ADB2Z5LU)** panel, modify the DBALIAS field as needed to specify the accelerators that you want, and press Enter.
7. If the **Statement Execution Prompt (ADB2PSTM)** panel is displayed, specify 1, and press Enter to execute the UPDATE statement.

Related tasks

"Defining an accelerator group" on page 931
Define an accelerator group when you want an accelerator-only table to be defined on more than one accelerator.

Starting and stopping accelerators

You start and stop an accelerator by using the Start Accelerator and Stop Accelerator panels.

About this task

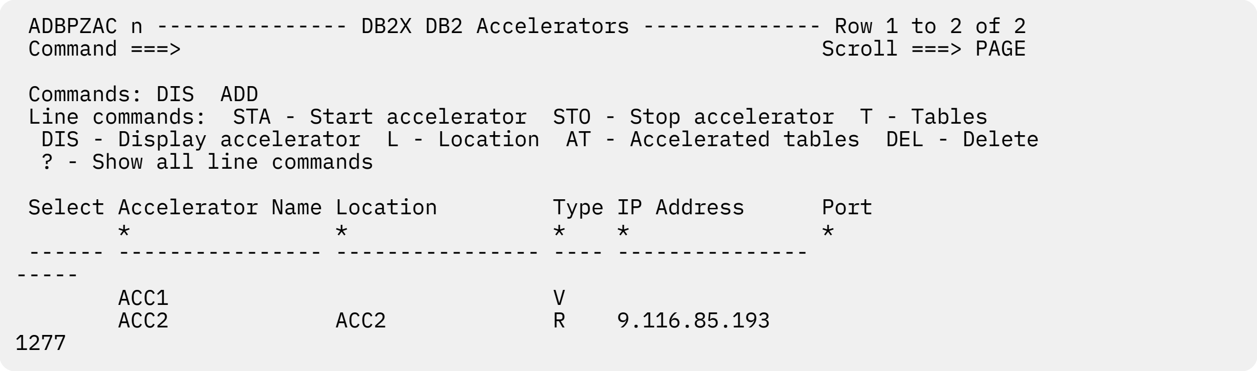
Before you can use an accelerator, you must start it. After you are done using an accelerator, you might want to stop it to conserve system resources. You might also want to stop an accelerator to terminate inactive accelerator threads.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed, as shown in the following figure.



```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ --------------- --------------- ---- ---------------
-----
        ACC1                              V
        ACC2            ACC2              R    9.116.85.193
1277
```

*Figure 449. **DB2 Accelerators (ADBPZAC)** panel*

3. Specify one of the following line commands:

Option	Description
STA	The DB2 Start Accelerator panel is displayed.
STO	The DB2 Stop Accelerator panel is displayed.

4. Specify the accelerator that you want to start or stop in the **Accelerator name** field. To start or stop all accelerators, Specify an asterisk (*). Press Enter.

Displaying accelerators

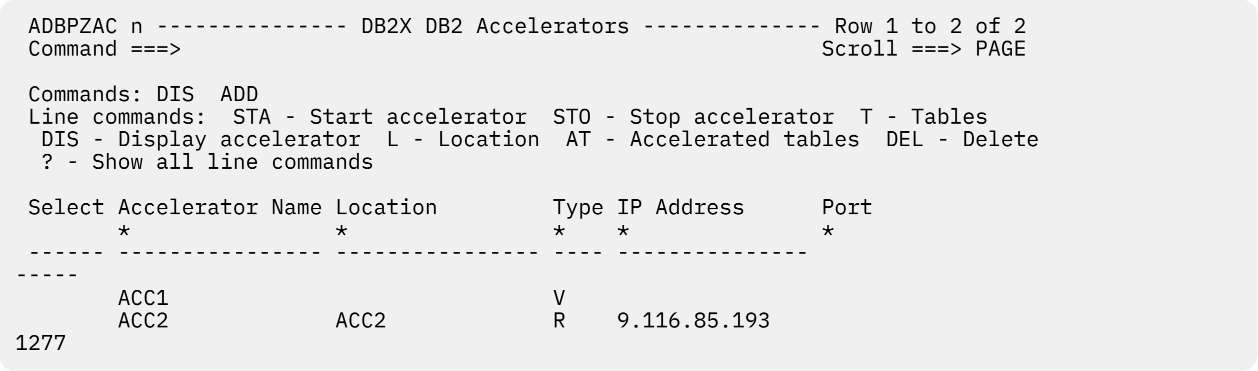
You can display information about the accelerators that are connected to your Db2 data server.

Procedure

To display accelerators:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel lists any accelerators:



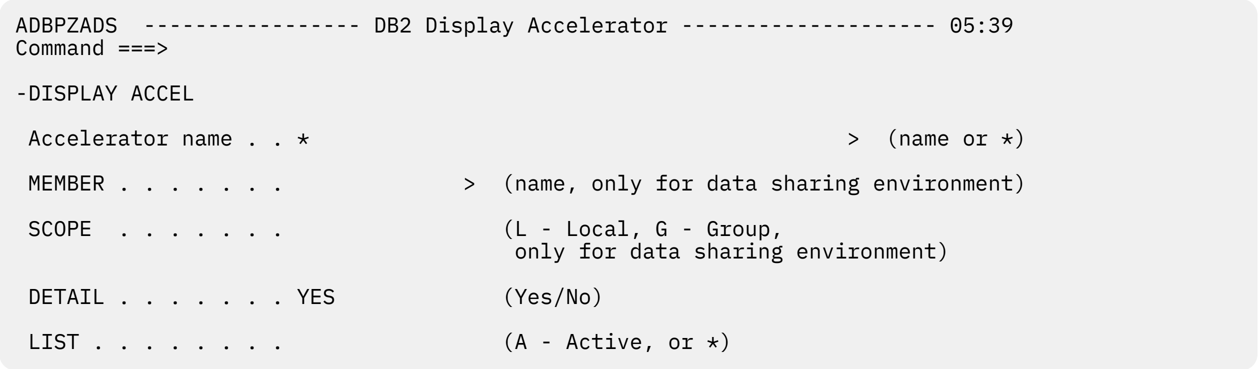
```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ ---------------- ---------------- ---- ---------------
-----
        ACC1                              V
        ACC2             ACC2             R    9.116.85.193
1277
```

   *Figure 450. **DB2 Accelerators (ADBPZAC)** panel*

   Note: For Db2 12 function level 509 or higher, this panel also lists accelerator groups (**Type** = G) and accelerator aliases (**Type** = A).

3. Specify the DIS primary command to display information about all of the accelerators or specify the DIS line command against an accelerator to display information about that particular accelerator, and press Enter.

   The **Display Accelerator (ADBPZADS)** panel is displayed:



```
ADBPZADS  ---------------- DB2 Display Accelerator ------------------- 05:39
Command ===>

-DISPLAY ACCEL

 Accelerator name . . *                                           >  (name or *)

 MEMBER . . . . . . .            >  (name, only for data sharing environment)

 SCOPE  . . . . . . .               (L - Local, G - Group,
                                     only for data sharing environment)

 DETAIL . . . . . . . YES           (Yes/No)

 LIST . . . . . . . .               (A - Active, or *)
```

   *Figure 451. **Display Accelerator (ADBPZADS)** panel*

   If you entered the DIS line command, the **Accelerator name** field contains the accelerator that you selected. If you entered the DIS primary command, the **Accelerator name** field contains an asterisk, indicating that all accelerators were selected.

4. Press Enter to view information about the accelerators.

Displaying accelerator status

You can display detailed information about the accelerators that are connected to your Db2 data server.

Procedure

To display accelerator status:

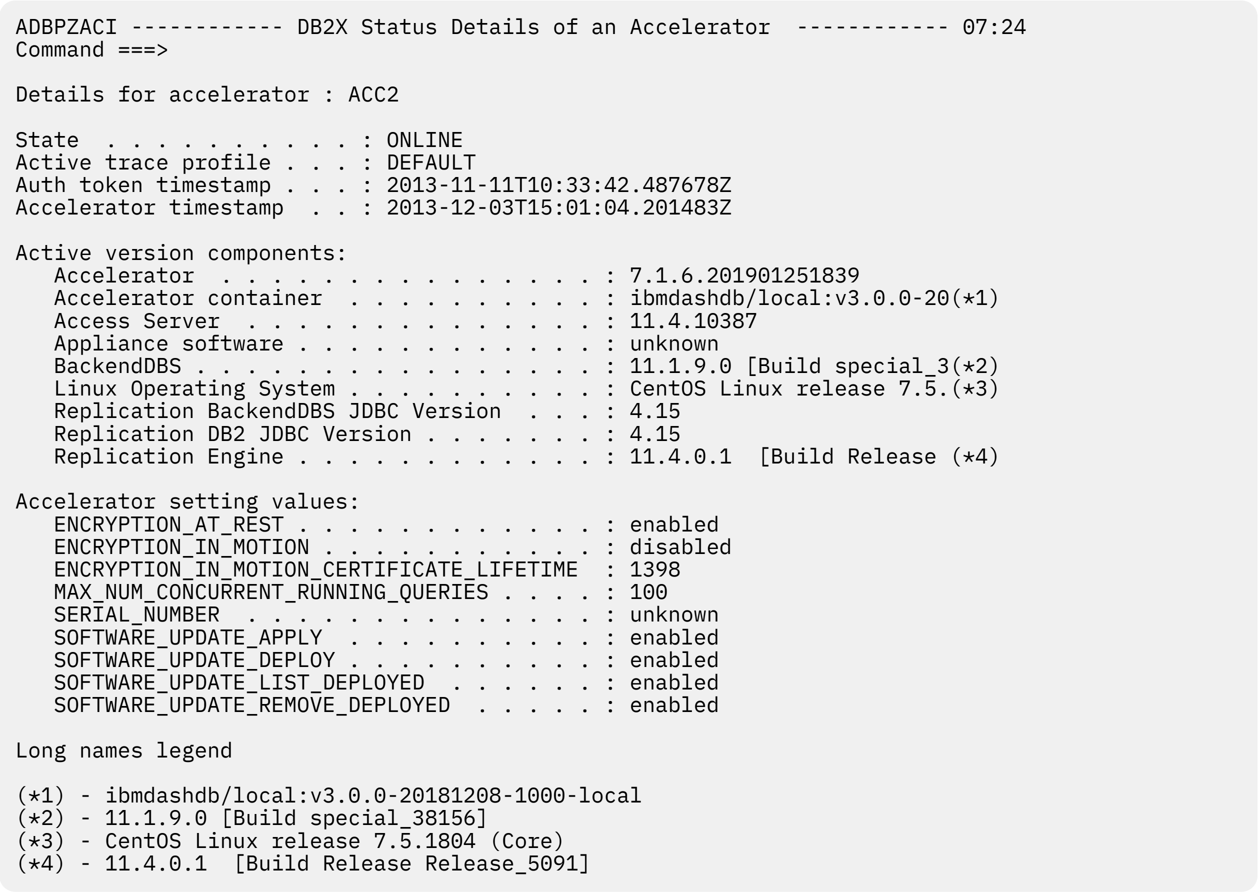
1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed:

```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ --------------- --------------- ---- ---------------
-----
        ACC1                              V
        ACC2             ACC2             R    9.116.85.193
1277
```

   The **DB2 Accelerators (ADBPZAC)** panel lists the accelerator servers that have been defined to Db2 and their locations. From this panel, you can perform various actions, including starting and stopping an accelerator server.
3. In the **Select** column, specify the I line command to view the accelerator status, and press Enter.

   The **Status details of an Accelerator (ADBPZACI)** panel is displayed:



```
ADBPZACI ------------ DB2X Status Details of an Accelerator  ------------ 07:24
Command ===>

Details for accelerator : ACC2

State  . . . . . . . . . . : ONLINE
Active trace profile . . . : DEFAULT
Auth token timestamp . . . : 2013-11-11T10:33:42.487678Z
Accelerator timestamp  . . : 2013-12-03T15:01:04.201483Z

Active version components:
   Accelerator  . . . . . . . . . . . . . . . : 7.1.6.201901251839
   Accelerator container  . . . . . . . . . . : ibmdashdb/local:v3.0.0-20(*1)
   Access Server  . . . . . . . . . . . . . . : 11.4.10387
   Appliance software . . . . . . . . . . . . : unknown
   BackendDBS . . . . . . . . . . . . . . . . : 11.1.9.0 [Build special_3(*2)
   Linux Operating System . . . . . . . . . . : CentOS Linux release 7.5.(*3)
   Replication BackendDBS JDBC Version  . . . : 4.15
   Replication DB2 JDBC Version . . . . . . . : 4.15
   Replication Engine . . . . . . . . . . . . : 11.4.0.1  [Build Release (*4)

Accelerator setting values:
   ENCRYPTION_AT_REST . . . . . . . . . . . . : enabled
   ENCRYPTION_IN_MOTION . . . . . . . . . . . : disabled
   ENCRYPTION_IN_MOTION_CERTIFICATE_LIFETIME  : 1398
   MAX_NUM_CONCURRENT_RUNNING_QUERIES . . . . : 100
   SERIAL_NUMBER  . . . . . . . . . . . . . . : unknown
   SOFTWARE_UPDATE_APPLY  . . . . . . . . . . : enabled
   SOFTWARE_UPDATE_DEPLOY . . . . . . . . . . : enabled
   SOFTWARE_UPDATE_LIST_DEPLOYED  . . . . . . : enabled
   SOFTWARE_UPDATE_REMOVE_DEPLOYED  . . . . . : enabled

Long names legend

(*1) - ibmdashdb/local:v3.0.0-20181208-1000-local
(*2) - 11.1.9.0 [Build special_38156]
(*3) - CentOS Linux release 7.5.1804 (Core)
(*4) - 11.4.0.1  [Build Release Release_5091]
```

Testing the accelerator connection

You can check whether a real accelerator is properly connected to the Db2 subsystem.

About this task

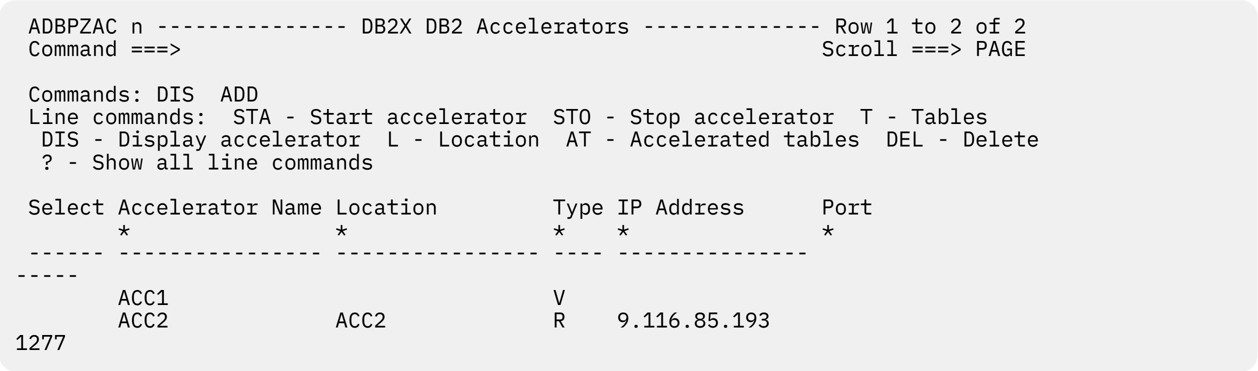
To verify that the accelerator is connected to Db2, you can run the CONN or PING commands.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed, as shown in the following figure.



```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ ---------------- ---------------- ---- ---------------
-----
        ACC1                              V
        ACC2             ACC2             R    9.116.85.193
1277
```

*Figure 452. **DB2 Accelerators (ADBPZAC)** panel*

3. Specify one of the following line commands, and press Enter.

Option	Description
CONN	Tests the DRDA connection between Db2 and accelerator
PING	Tests the IP connection between Db2 and accelerator

An informational message about your connection status will be displayed.

Displaying accelerator tasks

You can retrieve a list of tasks that are running on the accelerator. You can view task details and select or cancel tasks.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed, as shown in the following figure.

```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ ---------------- ---------------- ---- ---------------
-----
        ACC1                              V
        ACC2             ACC2             R    9.116.85.193
1277
```

*Figure 453. **DB2 Accelerators (ADBPZAC)** panel*

3. In the **Select** column, specify TASK beside the accelerator for which you want to retrieve task information.

   The **Accelerator Tasks panel (ADBPZA2)** panel is displayed, as shown in the following figure.

```
ADBPZA2 n --------------- DB2X Accelerator Tasks ------------ Row 1 to 10 of 10
Command ===>                                                    Scroll ===> CSR

Accelerator name: ACC1

Sel Id         Type             User      Pct    Age Status
    *          *                *           *      * *
--- ---------> ---------------> -------- ---- -----> -------------------------
>
    505542     LOAD             SYSADM    100      0 Collecting current task
.
.
.
```

*Figure 454. **Accelerator Tasks panel (ADBPZA2)** panel*

Canceling accelerator tasks

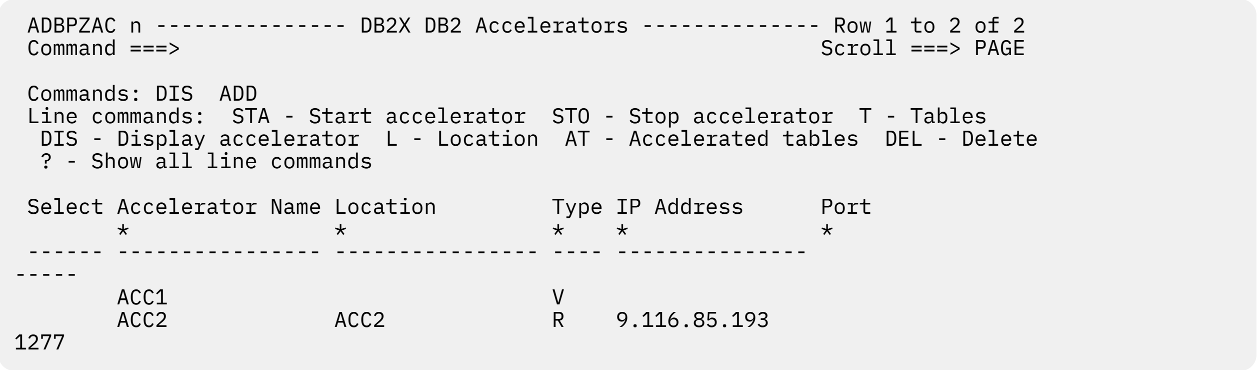
You can cancel tasks that are running on the accelerator.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed, as shown in the following figure.

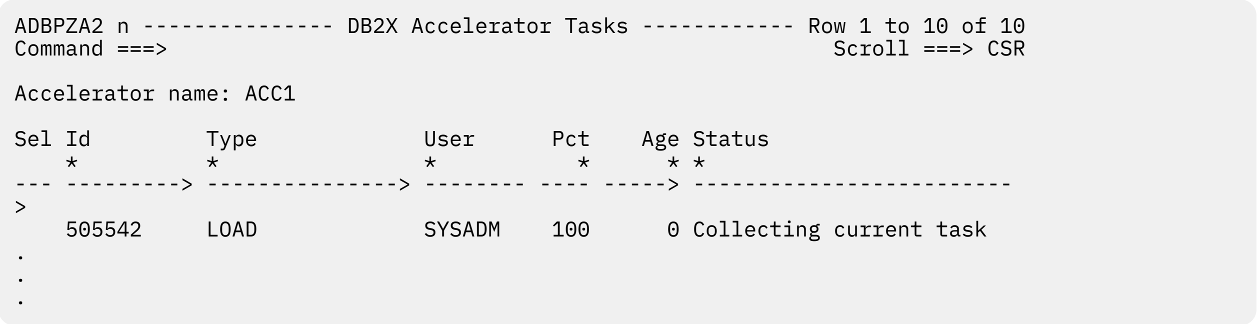


```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ ---------------- ---------------- ---- ---------------
-----
        ACC1                              V
        ACC2             ACC2             R    9.116.85.193
1277
```

*Figure 455. **DB2 Accelerators (ADBPZAC)** panel*

3. In the **Select** column, specify TASK beside the accelerator for which you want to retrieve task information.

   The **Accelerator Tasks panel (ADBPZA2)** panel is displayed, as shown in the following figure.



```
ADBPZA2 n --------------- DB2X Accelerator Tasks ------------ Row 1 to 10 of 10
Command ===>                                                    Scroll ===> CSR

Accelerator name: ACC1

Sel Id         Type             User      Pct    Age Status
    *          *                *           *      * *
--- ---------> ---------------> -------- ---- -----> -------------------------
>
    505542     LOAD             SYSADM    100      0 Collecting current task
.
.
.
```

*Figure 456. **Accelerator Tasks panel (ADBPZA2)** panel*

4. Specify CAN beside the task that you want to cancel, and press Enter.

Deleting accelerators

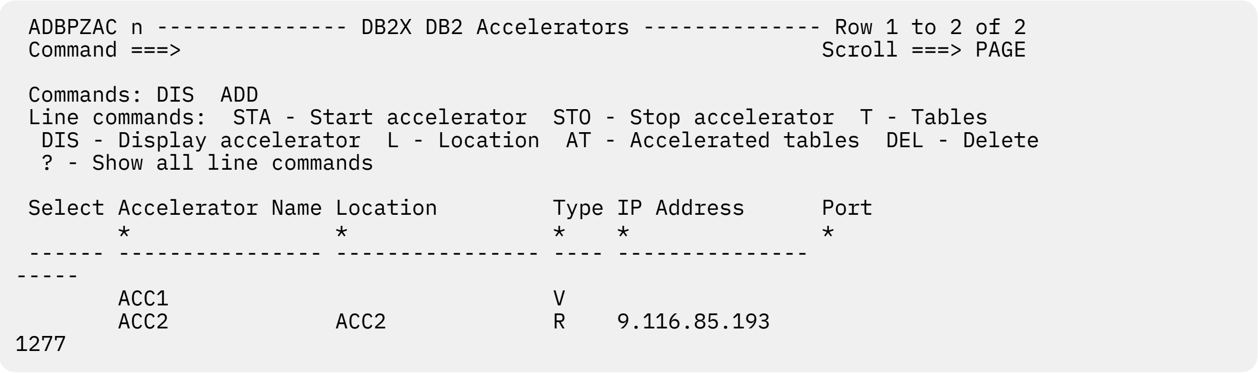
You delete accelerators by using the delete command.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed, as shown in the following figure.

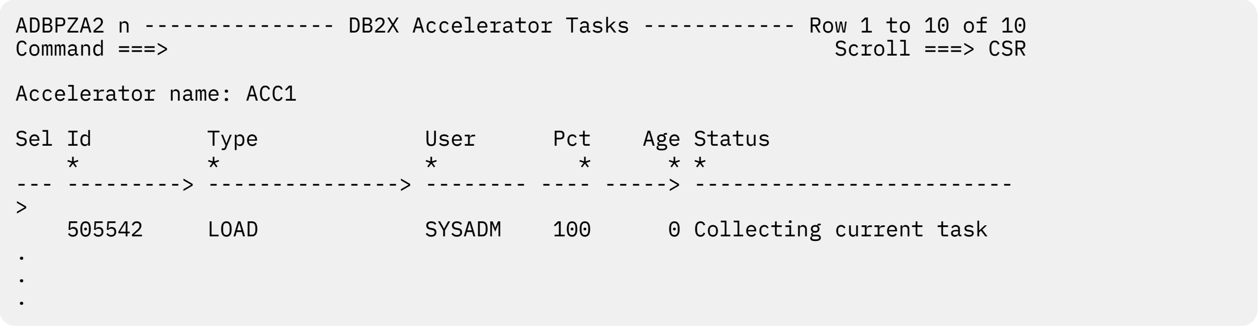


```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ ---------------- ---------------- ---- ---------------
-----
        ACC1                              V
        ACC2             ACC2             R    9.116.85.193
1277
```

Figure 457. ***DB2 Accelerators (ADBPZAC)*** *panel*

3. In the **Select** column, specify TASK beside the accelerator for which you want to retrieve task information.

   The **Accelerator Tasks panel (ADBPZA2)** panel is displayed, as shown in the following figure.

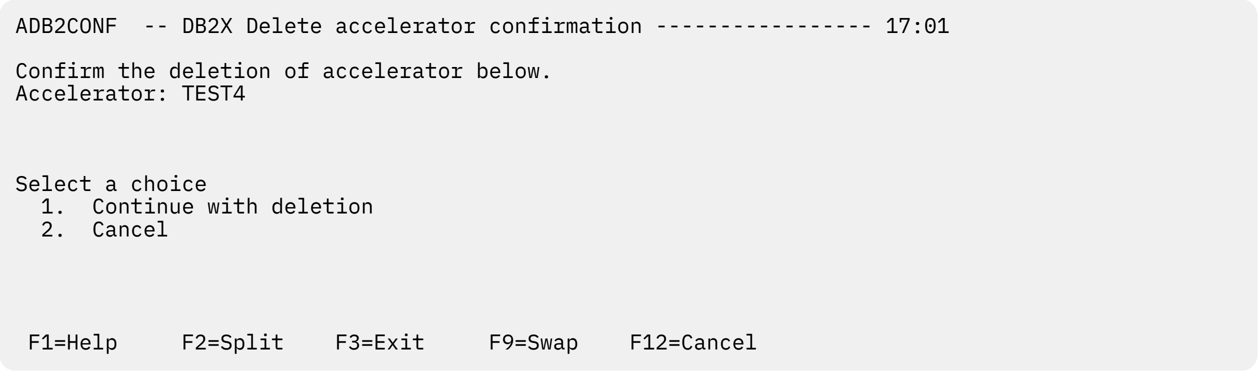


```
ADBPZA2 n --------------- DB2X Accelerator Tasks ------------ Row 1 to 10 of 10
Command ===>                                                   Scroll ===> CSR

Accelerator name: ACC1

Sel Id         Type             User      Pct    Age Status
    *          *                *           *      * *
--- ---------> ---------------> -------- ---- -----> ------------------------
>
    505542     LOAD             SYSADM    100      0 Collecting current task
.
.
.
```

Figure 458. ***Accelerator Tasks panel (ADBPZA2)*** *panel*

4. Specify DEL beside the accelerator that you want to delete, and press Enter.

   The **Delete Accelerator confirmation (ADB2CONF)** panel is displayed, as shown in the following figure.



```
ADB2CONF  -- DB2X Delete accelerator confirmation ----------------- 17:01

Confirm the deletion of accelerator below.
Accelerator: TEST4



Select a choice
  1.  Continue with deletion
  2.  Cancel




 F1=Help     F2=Split    F3=Exit     F9=Swap    F12=Cancel
```

Figure 459. ***Delete Accelerator confirmation (ADB2CONF)*** *panel*

5. Specify one of the following options, and press Enter.

Option	Description
1	Confirms the deletion.
2	Cancels the deletion.

   If you specify option 1, the following message is displayed:

```
Delete stmt executed
```

Creating trace profiles

You can create a trace profile for an accelerator. In a trace profile, you can edit trace level details and trace components.

About this task

Trace profiles provide the following information:

- The accelerator components to be traced and the trace detail level
- The approximate size of the compressed trace data
- Whether a trace record is immediately written to disk (flush) or if it is cached in memory for better performance

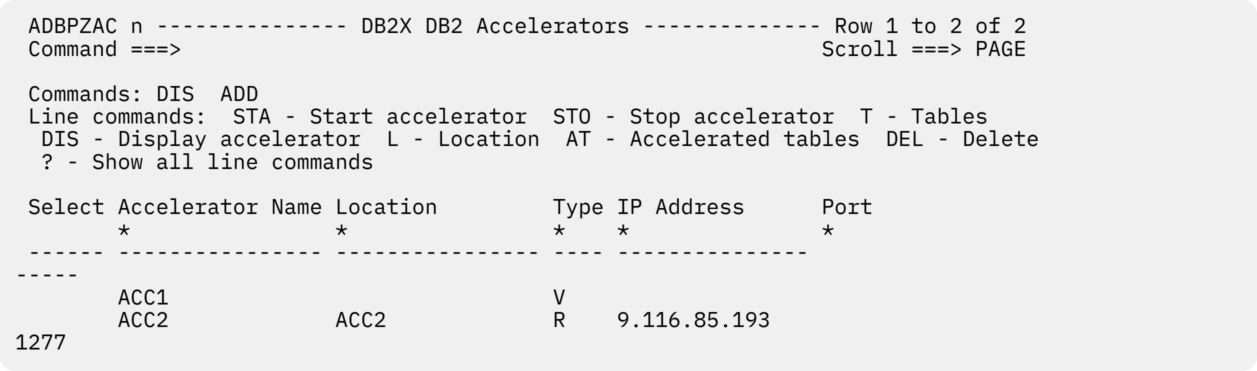
You can select and activate trace profiles. You can also define new profiles and remove existing ones. Several trace profiles are preconfigured on the accelerator.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed, as shown in the following figure.



```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ ---------------- ---------------- ---- ---------------
-----
        ACC1                              V
        ACC2             ACC2             R    9.116.85.193
1277
```

*Figure 460. **DB2 Accelerators (ADBPZAC)** panel*

3. In the **Select** column, specify TPR to display a list of trace profiles that are associated with the accelerator, and press Enter.

   The **Accelerator Trace Profiles (ADBPZTPR)** panel is displayed, as shown in the following figure.

```
ADBPZTPR  ----------- DB2X Accelerator Trace Profiles -------- Row 1 to 8 of 8
 Command ===>                                                  Scroll ===> PAGE

 Accelerator name : ACC1

 Commands: ADD
 Line commands:
  CO - Components   DESC - Description   DEL - Delete  ACT -
Activate

                                   Default         File  Force
 Sel  Name                  Active Level           Size  Flush
      *                     *      *                  * *
 ---- ---------------------- ------ -------------- ----- -----
      DEFAULT               YES    WARN               3 NO
      QUERY                 NO     WARN             160 NO
      LOAD                  NO     WARN              90 NO
      ARCHIVE               NO     WARN              90 NO
      REPLICATION           NO     WARN              63 NO
      SOFTWARE_UPDATE       NO     WARN              20 NO
      CONTROLLERS           NO     WARN              20 NO
      SYSTEM_CRASH          NO     TRACE           1000 YES
```

*Figure 461. **Accelerator Trace Profiles (ADBPZTPR)** panel*

4. Specify ADD to add a new trace profile, and press Enter.

   The **Add Accelerator Trace Profile (ADBPZTPA)** panel is displayed, as shown in the following figure.

```
ADBPZTPA ------ DB2X Add Accelerator Trace Profile ------------ 07:24
Command ===>

Commands: CONTINUE  COMP

Accelerator name . . ACC1
Profile name . . . . QUERY_NEW
Default level  . . .        (D-DEBUG, DE-DEBUG_EXTENDED, E-ERROR, F-FATAL,
                             I-INFO, O-OFF, T-TRACE or W-WARN)
File size in MB  . .        (1-4096)
Force flush  . . . .        (Yes/No)
Description  . . . .
```

*Figure 462. **Add Accelerator Trace Profile (ADBPZTPA)** panel*

5. Specify COMP command to define individual tracing components whose trace levels deviate from the default trace level, and press Enter.

   The **Accelerator Trace Profiles (ADBPZTPC)** panel is displayed, as shown in the following figure.

```
ADBPZTPC  ----------- DB2X Accelerator Trace Profiles ------------- 16:54
Command ===>                                             Scroll ===> PAGE

Accelerator name . . : ACC1
Trace profile name . : QUERY_NEW

Commands: CONTINUE  CANCEL

Line commands:
 I - Insert  R - Repeat  D - Delete

S    Name                           Level          Operation
     *                              *              *
---- ------------------------------ -------------- ----------
     ?                              ?
```

*Figure 463. **Accelerator Trace Profiles (ADBPZTPC)** panel*

   The following component names are valid: CATALOG, CONTROLLER, DRDA, PROFILING, REPLICATION, SERVICES or TRACE.

   The following component values are valid: DEBUG, ERROR, INFO, OFF, TRACE or WARN.

What to do next

You can activate a non-default or user-created trace profile by issuing the ACT command. Activating a trace profile allows you to capture and provide customized details to IBM Support.

Important: Using a trace profile other than the DEFAULT profile can lower the performance of the accelerator. Use a non-default profile only when instructed to do so by IBM Support. Re-enable the DEFAULT profile when the support activities are complete.

Displaying trace profiles

You can display the list of trace profiles that are associated with an accelerator. From the list, you can select a profile and modify its trace level details.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed, as shown in the following figure.

```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location        Type IP Address      Port
        *                *               *    *               *
 ------ --------------- --------------- ---- ---------------
-----
        ACC1                             V
        ACC2            ACC2             R    9.116.85.193
1277
```

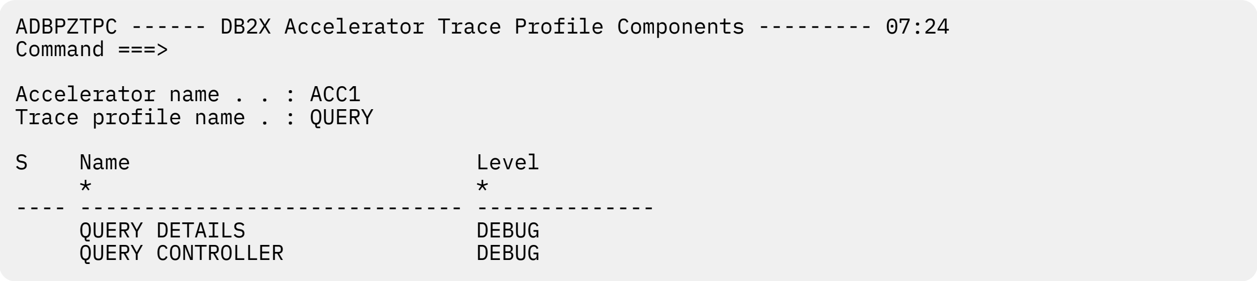
   Figure 464. ***DB2 Accelerators (ADBPZAC)*** *panel*
3. In the **Select** column, specify TPR to display a list of trace profiles associated with the accelerator, and press Enter.

   The **Accelerator Trace Profiles (ADBPZTPR)** panel is displayed, as shown in the following figure.

```
ADBPZTPR  ----------- DB2X Accelerator Trace Profiles -------- Row 1 to 8 of 8
 Command ===>                                                  Scroll ===> PAGE

 Accelerator name : ACC1

 Commands: ADD
 Line commands:
  CO - Components   DESC - Description   DEL - Delete  ACT -
Activate

                                   Default         File  Force
 Sel  Name                   Active Level           Size  Flush
      *                      *      *                  * *
 ---- ---------------------- ------ -------------- ----- -----
      DEFAULT                YES    WARN               3 NO
      QUERY                  NO     WARN             160 NO
      LOAD                   NO     WARN              90 NO
      ARCHIVE                NO     WARN              90 NO
      REPLICATION            NO     WARN              63 NO
      SOFTWARE_UPDATE        NO     WARN              20 NO
      CONTROLLERS            NO     WARN              20 NO
      SYSTEM_CRASH           NO     TRACE           1000 YES
```

   Figure 465. ***Accelerator Trace Profiles (ADBPZTPR)*** *panel*
4. Specify CO to display the trace profile components, and press Enter.

The **Accelerator Trace Profile Components (ADBPZTPC)** panel is displayed, as shown in the following figure.



```
ADBPZTPC ------ DB2X Accelerator Trace Profile Components --------- 07:24
Command ===>

Accelerator name . . : ACC1
Trace profile name . : QUERY

S    Name                           Level
     *                              *
---- ------------------------------ --------------
     QUERY DETAILS                  DEBUG
     QUERY CONTROLLER               DEBUG
```

Figure 466. ***Accelerator Trace Profile Components (ADBPZTPC)*** *panel*

5. Specify DESC to view a description of the selected trace profile, and press Enter.

Deleting trace profiles

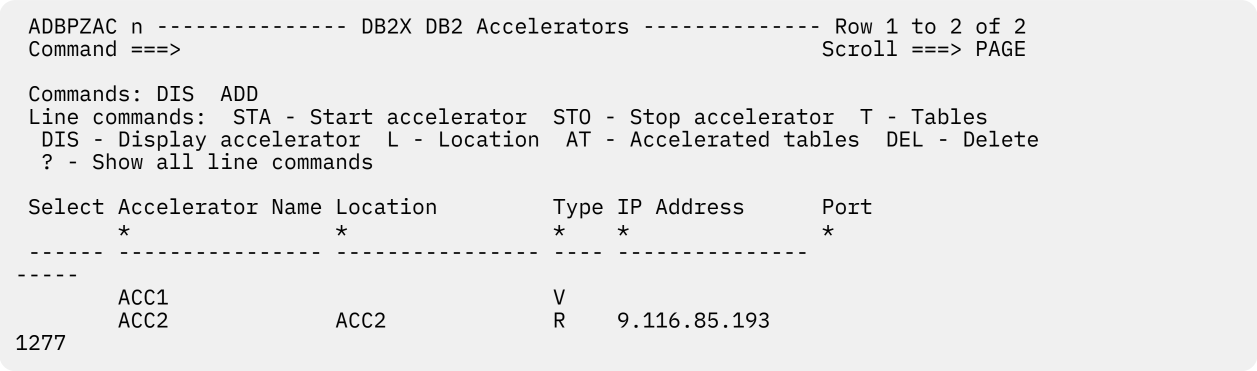
You can delete trace profiles by using the delete command.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed, as shown in the following figure.

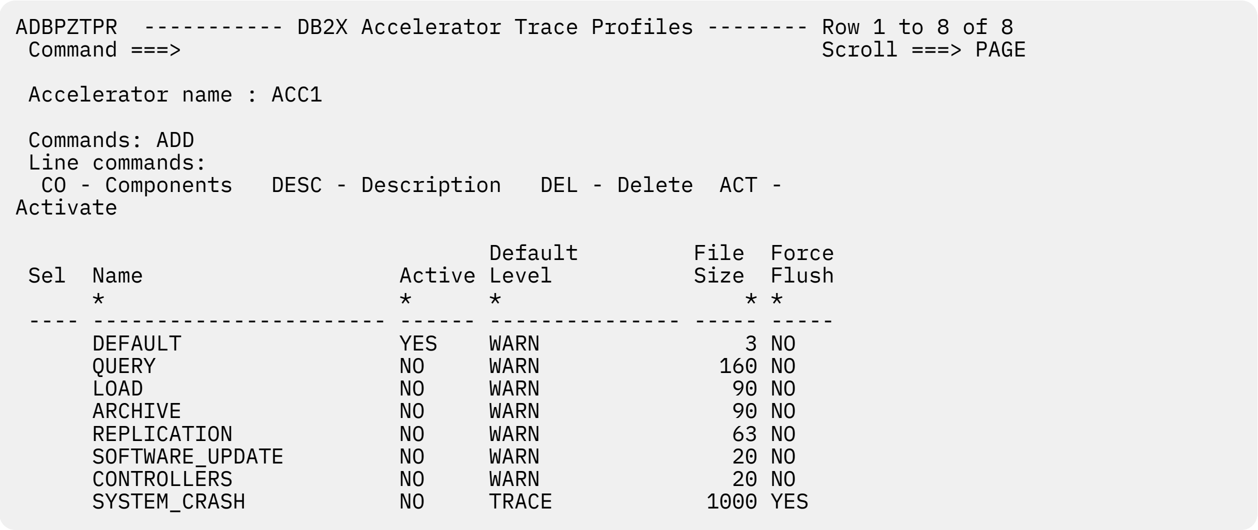


```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ ---------------- ---------------- ---- ---------------
-----
        ACC1                              V
        ACC2             ACC2             R    9.116.85.193
1277
```

Figure 467. ***DB2 Accelerators (ADBPZAC)*** *panel*

3. In the **Select** column, specify TPR to display a list of trace profiles associated with the accelerator, and press Enter.

   The **Accelerator Trace Profiles (ADBPZTPR)** panel is displayed, as shown in the following figure.



```
ADBPZTPR  ----------- DB2X Accelerator Trace Profiles -------- Row 1 to 8 of 8
 Command ===>                                                Scroll ===> PAGE

 Accelerator name : ACC1

 Commands: ADD
 Line commands:
  CO - Components   DESC - Description   DEL - Delete  ACT -
Activate

                                     Default         File  Force
 Sel  Name                    Active Level           Size  Flush
      *                       *      *                  * *
 ---- ---------------------- ------ -------------- ----- -----
      DEFAULT                 YES    WARN               3 NO
      QUERY                   NO     WARN             160 NO
      LOAD                    NO     WARN              90 NO
      ARCHIVE                 NO     WARN              90 NO
      REPLICATION             NO     WARN              63 NO
      SOFTWARE_UPDATE         NO     WARN              20 NO
      CONTROLLERS             NO     WARN              20 NO
      SYSTEM_CRASH            NO     TRACE           1000 YES
```

*Figure 468. **Accelerator Trace Profiles (ADBPZTPR)** panel*

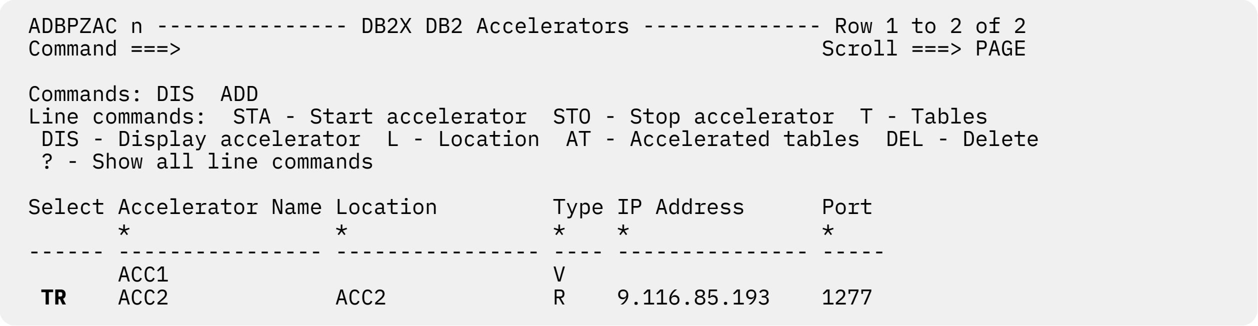
4. Specify DEL beside the trace profile that you want to delete, and press Enter.

Retrieving trace data for accelerators

You can retrieve trace data about the accelerators that are connected to your Db2 data server. Trace data contains information about system status, database runtime, query execution plans, and catalog dumps. You can use this information to diagnose accelerator problems.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AC, and press Enter.
3. On the **DB2 Accelerators (ADBPZAC)** panel, in the **Select** column, specify TR beside the accelerator for which you want to retrieve trace data, and press Enter:

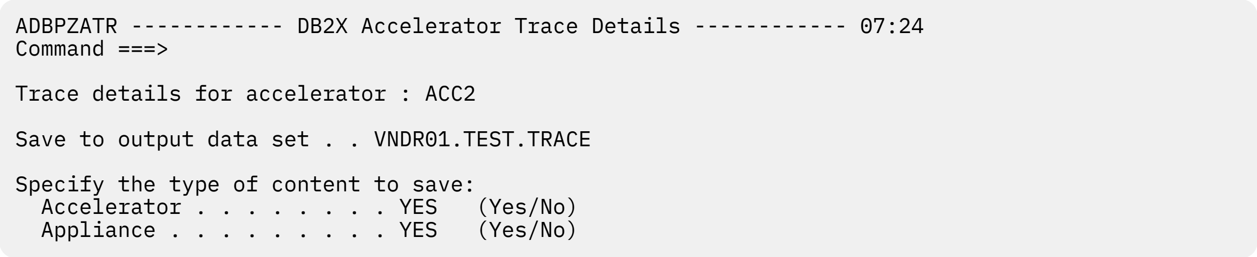


```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location         Type IP Address      Port
        *                *                *    *               *
 ------ ---------------- ---------------- ---- --------------- -----
        ACC1                              V
  TR    ACC2             ACC2             R    9.116.85.193    1277
```

*Figure 469. **DB2 Accelerators (ADBPZAC)** panel*

4. On the **DB2 Accelerator Trace Details (ADBPZATR)** panel, specify the type of trace data to save to the specified sequential output data set, and press Enter.

   This panel is displayed as follows:



```
ADBPZATR ------------ DB2X Accelerator Trace Details ------------ 07:24
Command ===>

Trace details for accelerator : ACC2

Save to output data set . . VNDR01.TEST.TRACE

Specify the type of content to save:
  Accelerator . . . . . . . . YES   (Yes/No)
  Appliance . . . . . . . . . YES   (Yes/No)
```

Figure 470. ***DB2 Accelerator Trace Details (ADBPZATR)*** *panel*

After you press Enter, trace data is generated in the specified data set.

Updating accelerator credentials

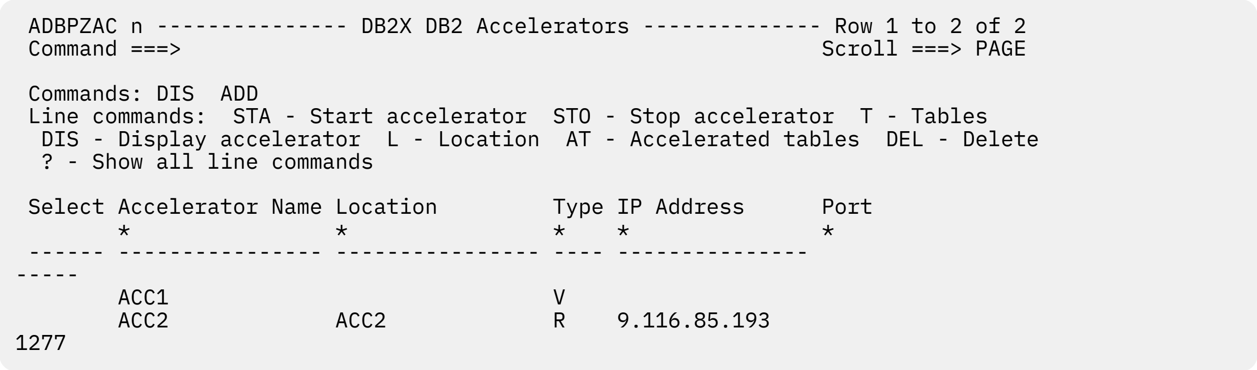
You can update the authentication token associated with the accelerator. You may want to update the authentication token periodically to ensure security.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option AC, and press Enter.

   The **DB2 Accelerators (ADBPZAC)** panel is displayed, as shown in the following figure.



```
 ADBPZAC n --------------- DB2X DB2 Accelerators -------------- Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> PAGE

 Commands: DIS  ADD
 Line commands:  STA - Start accelerator  STO - Stop accelerator  T - Tables
  DIS - Display accelerator  L - Location  AT - Accelerated tables  DEL - Delete
  ? - Show all line commands

 Select Accelerator Name Location        Type IP Address      Port
        *                *               *    *               *
 ------ --------------- --------------- ---- ---------------
-----
        ACC1                             V
        ACC2             ACC2            R    9.116.85.193
1277
```

Figure 471. ***DB2 Accelerators (ADBPZAC)*** *panel*

3. In the **Select** column, specify UC beside the accelerator whose credentials you want to update the accelerator credentials, and press Enter.

Accelerated table management

An accelerated table is a Db2 table that is added to an accelerator for the purpose of query enhancement. You use Db2 Admin Tool to add, display, load, enable, disable, archive, restore archive, and delete accelerated tables. You can also view the status details of accelerated tables and control the automatic reloading or removal of accelerated tables.

Information about the accelerated tables is stored in the pseudo-catalog table, SYSACCEL.SYSACCELERATEDTABLES. Each Db2 connection instance has one pseudo-catalog table.

Adding accelerator-shadow tables

You must add a table to the accelerator before you can use the accelerator for query enhancement or storage saver purposes. An *accelerator-shadow table* exists both in Db2 and in the accelerator.

Before you begin

You cannot add a Db2 table to an accelerator if any of the following conditions are true:

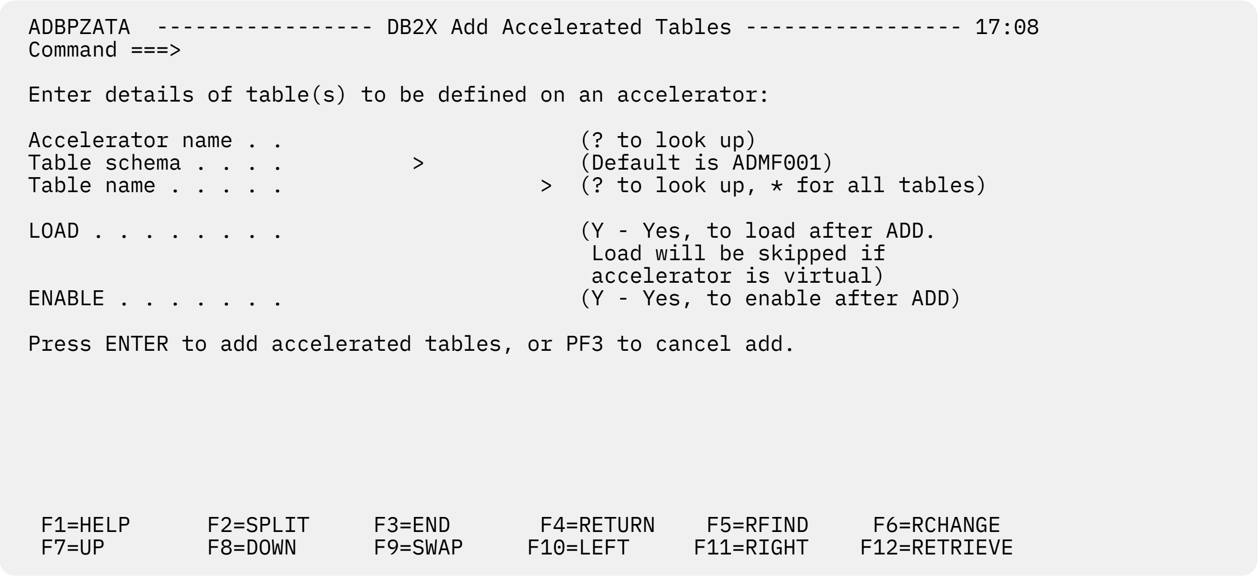
- The table is not a base table; that is, the value in the TYPE column of the SYSIBM.SYSTABLES table is not T.
- The table uses a row-level security label; that is, the value in the SECURITY_LABEL column of the SYSIBM.SYSTABLES table is R.
- The row-level access control is defined for the table; that is, the value in the CONTROL column of the SYSIBM.SYSTABLES table is R or B.

Additional conditions can prevent tables from being added to an accelerator. For a complete list of restrictions, see SYSPROC.ACCEL_ADD_TABLES

Procedure

To add an accelerator-shadow table:

1. Navigate to the **Add Accelerated Tables (ADBPZATA)** panel by taking one of the following paths:
   - To add a table from the accelerated table panels, complete the following steps:
     a. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
     b. On the **System Administration (ADB2Z)** panel, specify option AT , and press Enter.
     c. On the **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel, specify option 1, and press Enter.
     d. On the **Display Accelerated Tables (ADBPZAT)** panel, specify ADD, and press Enter.
   - To add a table from the system catalog, complete the following steps:
     a. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
     b. On the **System Catalog (ADB21)** panel, specify T, and press Enter.
     c. On the **Tables, Views, and Aliases (ADB21T)** panel, specify ADDA next to the table that you want to add, and press Enter.
2. On the **Add Accelerated Tables (ADBPZATA)** panel, specify the appropriate information to define a Db2 table on an accelerator.



```
ADBPZATA  ----------------- DB2X Add Accelerated Tables ----------------- 17:08
Command ===>

Enter details of table(s) to be defined on an accelerator:

Accelerator name . .                       (? to look up)
Table schema . . . .          >            (Default is ADMF001)
Table name . . . . .                    >  (? to look up, * for all tables)

LOAD . . . . . . . .                       (Y - Yes, to load after ADD.
                                            Load will be skipped if
                                            accelerator is virtual)
ENABLE . . . . . . .                       (Y - Yes, to enable after ADD)

Press ENTER to add accelerated tables, or PF3 to cancel add.







 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 472. Add Accelerated Tables panel (ADBPZATA)

Complete the following fields:

Accelerator Name
The name of the accelerator on which the table is to be defined.

Table schema
The schema of the table to be defined.

Table name
The name of the table to be defined.

LOAD
Specify whether to load the table in the accelerator by using the LOAD stored procedure. The table is not loaded if the accelerator is a virtual accelerator.

ENABLE
Specify whether to enable the table for acceleration. If the accelerator is real, the SYSPROC.ACCEL_SET_TABLES_ACCELERATION stored procedure is called. If the accelerator is virtual, the ENABLE column in SYSACCELERATEDTABLES is set to Y. If both LOAD and ENABLE are requested, LOAD is performed first, followed by ENABLE.

Information about selected Db2 table is inserted into the SYSACCEL.SYSACCELERATEDTABLES table.

Related information

Types of accelerator tables (Db2 12 for z/OS documentation)

Adding accelerator-only tables

An *accelerator-only table* exists only on an accelerator.

Before you begin

Adding an accelerator-only table to an accelerator group or alias requires Db2 12 function level 509 or higher and IBM Db2 Analytics Accelerator for z/OS (IDAA) 7.5.4 or higher.

Procedure

To add an accelerator-only table:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 2, and press Enter.
2. On the **Execute SQL Statements (ADB22)** panel, specify option 4, and press Enter.
3. On the **Create Table (ADB26CT)** panel, specify parameter values, and press Enter.

For help with the parameters on this panel, see “Creating tables” on page 334.

4. On the **Create Table Columns (ADB26CTF)** panel, specify column information and an accelerator:

```
ADB26CTF  ------------------ DD1A Create Table Columns ------- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> CSR

Schema  . .          >            Database  . . .
Name  . . . NEWTABLE          >  Table space . .
                                  Accelerator . . ACTESTAS

Commands :  CREATE  PRIMKEY  TBLOPTS  LONGNAMES  PART  HASH
Line commands:   M - Move  A - After  B - Before
 Inn - Insert  U - Update  D - Delete  Rnn - Repeat
 UM - Update XML modifiers
                                                                Operation
Select Column Name        Col Type  Length Scale  Null D Col No Type
       *                  *              *      * *    *      * *
------ -----------------> --------- ------ ------ ---- - ------ ---------

*      T1                 TIMESTMP      13     11 N    N      1 UPDATE
*      T2                 TIMESTZ       15     11 N    N      2 UPDATE
*      T3                 TIMESTZ       12      6 N    N      3 UPDATE
*      T4                 DATE           4      0 N    N      4 UPDATE
*      T5                 INTEGER        4      0 N    N      5 UPDATE
*      T6                 DATE           4      0 N    N      6 UPDATE
******************************* END OF DB2 DATA ******************************
```

Tip: In the **Accelerator** field, use the look up feature (?) to look up an accelerator name or group. The resulting **DB2 Accelerators (ADBPZAC)** panel lists the available accelerators and groups. The **Type** column indicates whether the accelerator is real (R), virtual (V), a group (G), or an alias (A). Select one by typing + in the **Select** column.

5. Issue the CREATE command, and press Enter.

Results

The CREATE TABLE statement with an IN ACCELERATOR clause is executed and the accelerator-only table is created.

Related information

Types of accelerator tables (Db2 12 for z/OS documentation)

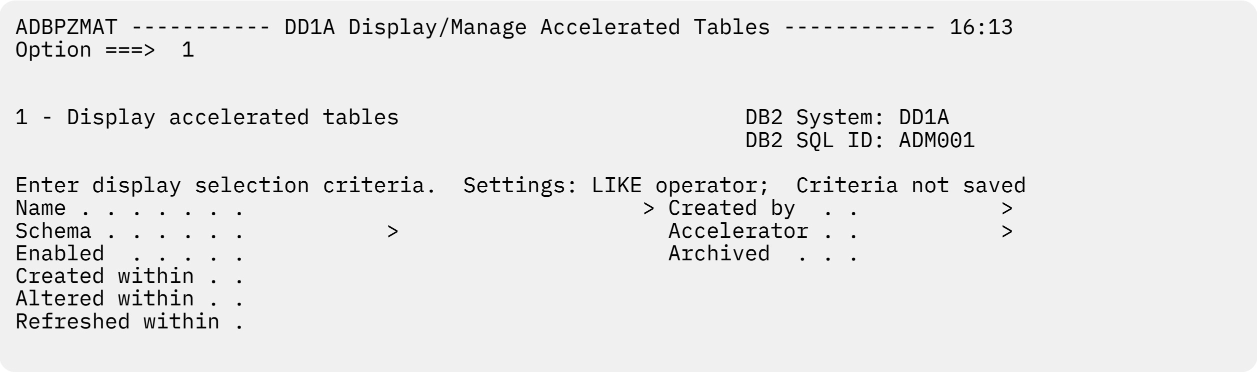
Displaying accelerator tables

You can display information about the tables that are associated with an accelerator, including both accelerator-shadow tables and accelerator-only tables.

Procedure

To display accelerated tables:

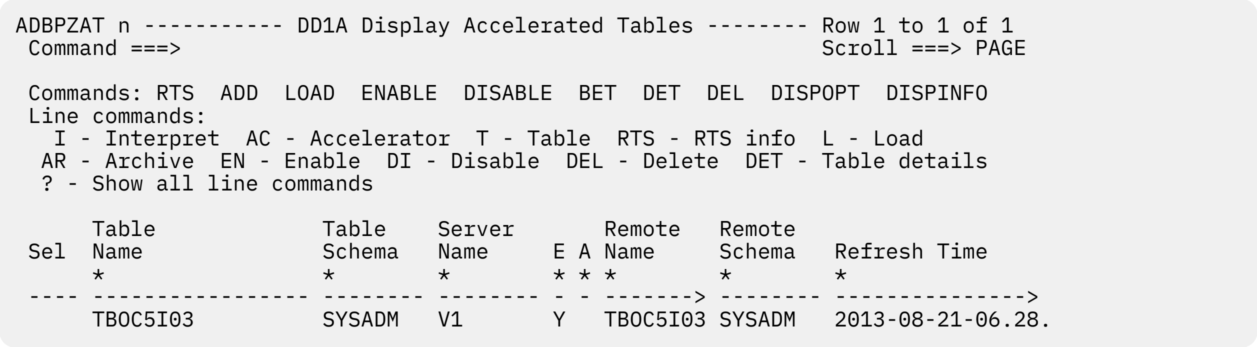
1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AT, and press Enter.
3. On the **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel optionally specify any filtering selection criteria. Then specify option 1, and press Enter.



```
ADBPZMAT ----------- DD1A Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                           DB2 System: DD1A
                                                         DB2 SQL ID: ADM001

Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                               > Created by  . .           >
Schema . . . . . .           >                     Accelerator . .           >
Enabled  . . . . .                                 Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

*Figure 473. **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel*

The **Display Accelerated Tables (ADBPZAT)** panel is displayed:



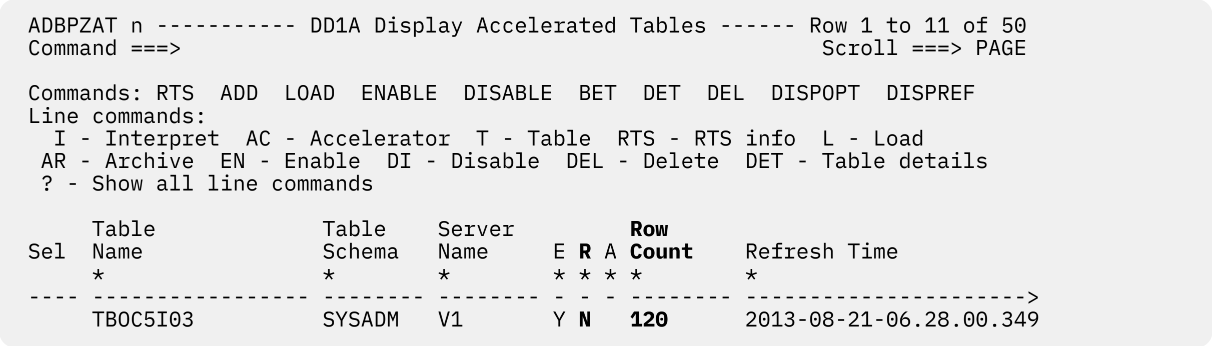
```
ADBPZAT n ----------- DD1A Display Accelerated Tables -------- Row 1 to 1 of 1
 Command ===>                                                  Scroll ===> PAGE

 Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPINFO
 Line commands:
   I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
  AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
  ? - Show all line commands

      Table             Table    Server       Remote   Remote
 Sel  Name              Schema   Name     E A Name     Schema   Refresh Time
      *                 *        *        * * *        *        *
 ---- ----------------- -------- -------- - - -------> -------- --------------->
      TBOC5I03          SYSADM   V1       Y   TBOC5I03 SYSADM   2013-08-21-06.28.
```

*Figure 474. **Display Accelerated Tables (ADBPZAT)** panel*

4. Use the following commands to display additional information:

DISPINFO

This primary command displays replication status and row count on the panel, as follows:



```
ADBPZAT n ----------- DD1A Display Accelerated Tables ------ Row 1 to 11 of 50
Command ===>                                                  Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPREF
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
 ? - Show all line commands

     Table             Table    Server         Row
Sel  Name              Schema   Name     E R A Count    Refresh Time
     *                 *        *        * * * *        *
---- ----------------- -------- -------- - - - -------- ---------------------->
     TBOC5I03          SYSADM   V1       Y N   120      2013-08-21-06.28.00.349
```

*Figure 475. **Display Accelerated Tables (ADBPZAT)** panel after DISPINFO command is issued*

To restore the original display, specify REFRESH or DISPREF.

DET

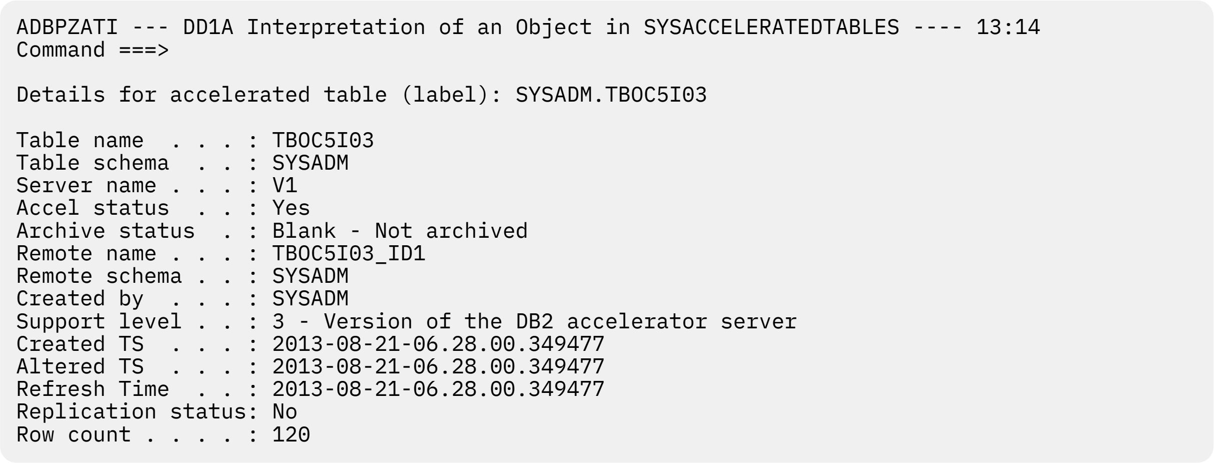
This line command generates a detail report online for the accelerated table.

BET

This line command generates a batch job that will create a detail report for the accelerated table.

I

This line command displays details for the accelerated table on the **Interpretation of an Object in SYSACCELERATEDTABLES (ADBPZATI)** panel:



```
ADBPZATI --- DD1A Interpretation of an Object in SYSACCELERATEDTABLES ---- 13:14
Command ===>

Details for accelerated table (label): SYSADM.TBOC5I03

Table name  . . . : TBOC5I03
Table schema  . . : SYSADM
Server name . . . : V1
Accel status  . . : Yes
Archive status  . : Blank - Not archived
Remote name . . . : TBOC5I03_ID1
Remote schema . . : SYSADM
Created by  . . . : SYSADM
Support level . . : 3 - Version of the DB2 accelerator server
Created TS  . . . : 2013-08-21-06.28.00.349477
Altered TS  . . . : 2013-08-21-06.28.00.349477
Refresh Time  . . : 2013-08-21-06.28.00.349477
Replication status: No
Row count . . . . : 120
```

Figure 476. ***Interpretation of an Object in SYSACCELERATEDTABLES (ADBPZATI)*** *panel*

Related information

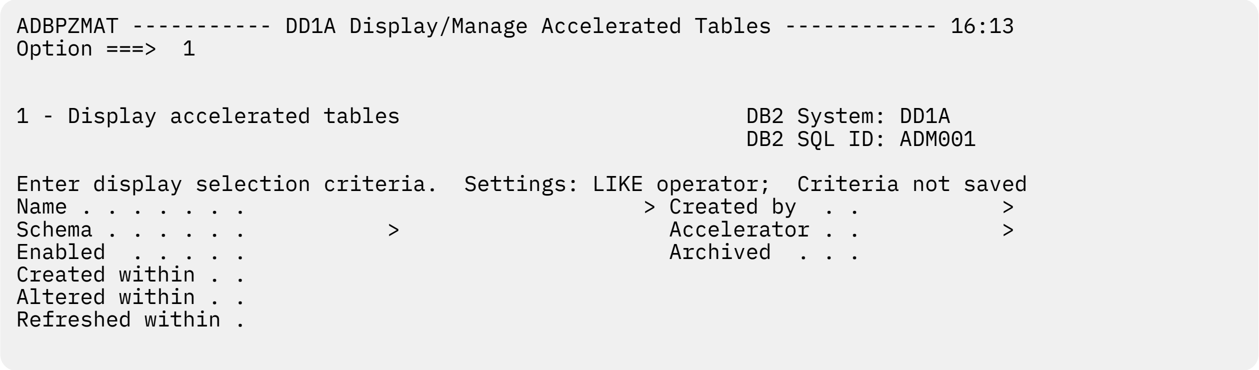
Types of accelerator tables (Db2 12 for z/OS documentation)

Loading accelerated tables

You can load Db2 table data into the accelerator after its definition has been copied to the accelerator.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AT, and press Enter.
3. On the **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel, specify any filters for the tables you want to display by entering selection criteria:



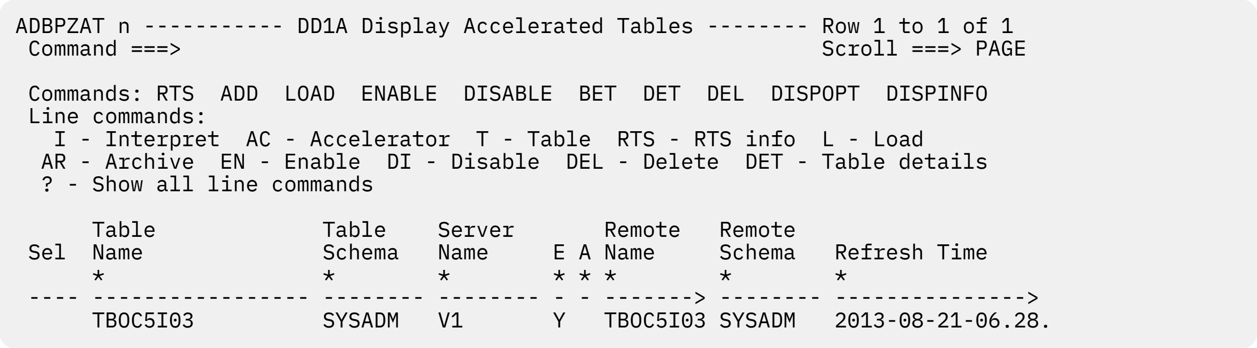
```
ADBPZMAT ----------- DD1A Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                           DB2 System: DD1A
                                                         DB2 SQL ID: ADM001

Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                                 > Created by  . .           >
Schema . . . . . .            >                      Accelerator . .           >
Enabled  . . . . .                                   Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

Figure 477. ***DB2 Display/Manage Accelerated Tables (ADBPZMAT)*** *panel*

4. Specify option 1, and press Enter.
5. On the **Display Accelerated Tables (ADBPZAT)** panel, specify the L line command to load the data of a selected table.

   The LOAD primary command loads the data of all of the selected tables.

   Note: You can load the table data to the accelerator in batch to reduce wait time. To run batch, specify YES for **Run Accelerator functions in batch** on the **Admin Defaults (ADB2P2)** panel.



```
ADBPZAT n ----------- DD1A Display Accelerated Tables -------- Row 1 to 1 of 1
 Command ===>                                                  Scroll ===> PAGE

 Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPINFO
 Line commands:
   I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
  AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
  ? - Show all line commands

      Table             Table    Server       Remote   Remote
 Sel  Name              Schema   Name     E A Name     Schema   Refresh Time
      *                 *        *        * * *        *        *
 ---- ----------------- -------- -------- - - -------> -------- --------------->
      TBOC5I03          SYSADM   V1       Y   TBOC5I03 SYSADM   2013-08-21-06.28.
```

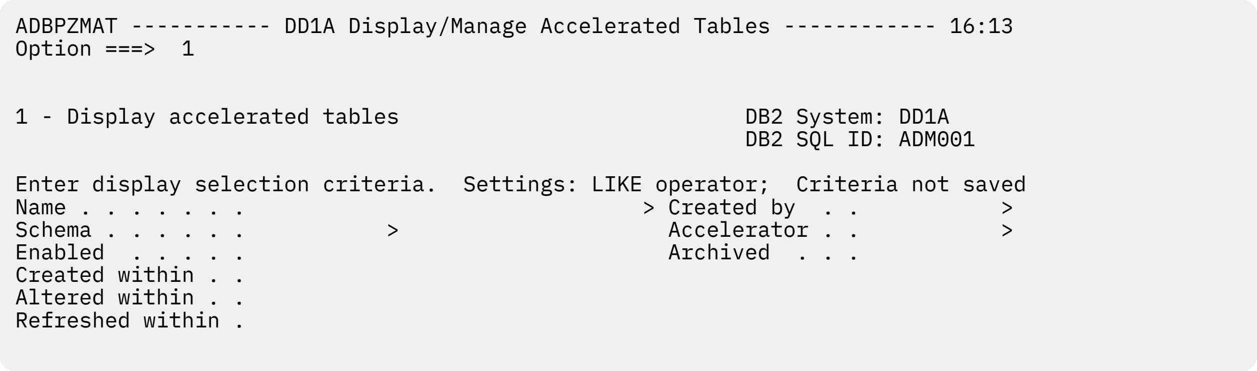
*Figure 478. **Display Accelerated Tables (ADBPZAT)** panel*

Enabling and disabling accelerated tables

You can enable or disable an accelerated table to enable or disable query offloading for that Db2 table.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AT, and press Enter.
3. On the **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel, specify any filters for the tables you want to display by entering selection criteria:

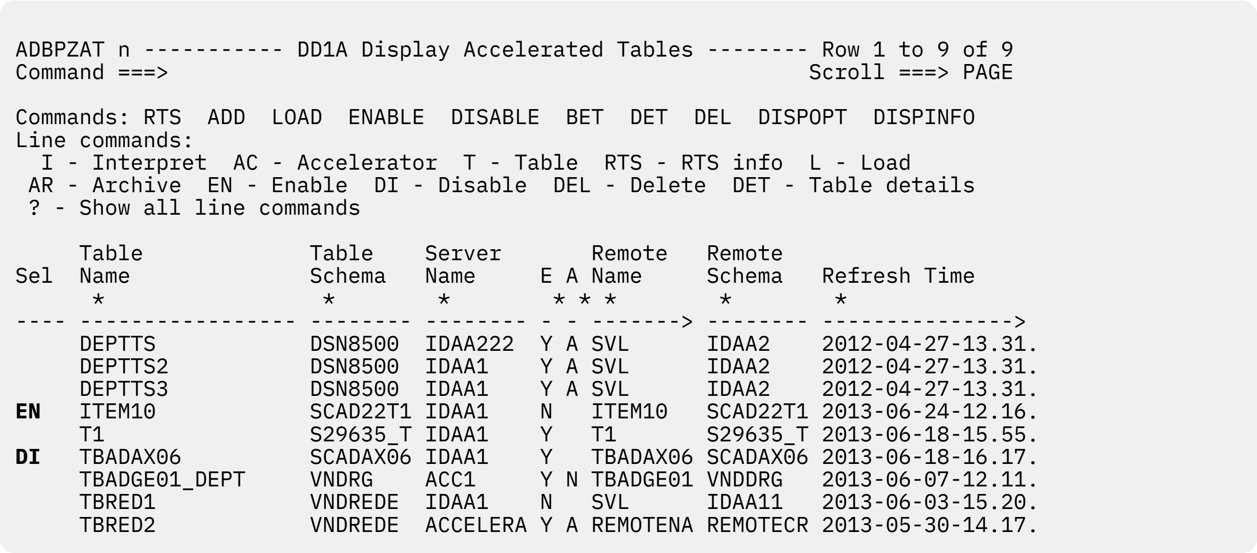


```
ADBPZMAT ----------- DD1A Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                           DB2 System: DD1A
                                                         DB2 SQL ID: ADM001

Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                               > Created by  . .           >
Schema . . . . . .           >                     Accelerator . .           >
Enabled  . . . . .                                 Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

*Figure 479. **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel*

4. Specify option 1, and press Enter.
5. On the **Display Accelerated Tables (ADBPZAT)** panel, specify the EN line command to enable an accelerator or the DI line command to disable an accelerator, and press Enter.

   The example in the following figure shows both the EN line command and DI line command specified:



```
ADBPZAT n ----------- DD1A Display Accelerated Tables -------- Row 1 to 9 of 9
Command ===>                                                  Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPINFO
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
 ? - Show all line commands

     Table              Table    Server       Remote   Remote
Sel  Name               Schema   Name     E A Name     Schema   Refresh Time
      *                  *        *        * * *        *        *
---- ------------------ -------- -------- - - -------> -------- --------------->
     DEPTTS             DSN8500  IDAA222  Y A SVL      IDAA2    2012-04-27-13.31.
     DEPTTS2            DSN8500  IDAA1    Y A SVL      IDAA2    2012-04-27-13.31.
     DEPTTS3            DSN8500  IDAA1    Y A SVL      IDAA2    2012-04-27-13.31.
EN   ITEM10             SCAD22T1 IDAA1    N   ITEM10   SCAD22T1 2013-06-24-12.16.
     T1                 S29635_T IDAA1    Y   T1       S29635_T 2013-06-18-15.55.
DI   TBADAX06           SCADAX06 IDAA1    Y   TBADAX06 SCADAX06 2013-06-18-16.17.
     TBADGE01_DEPT      VNDRG    ACC1     Y N TBADGE01 VNDDRG   2013-06-07-12.11.
     TBRED1             VNDREDE  IDAA1    N   SVL      IDAA11   2013-06-03-15.20.
     TBRED2             VNDREDE  ACCELERA Y A REMOTENA REMOTECR 2013-05-30-14.17.
```

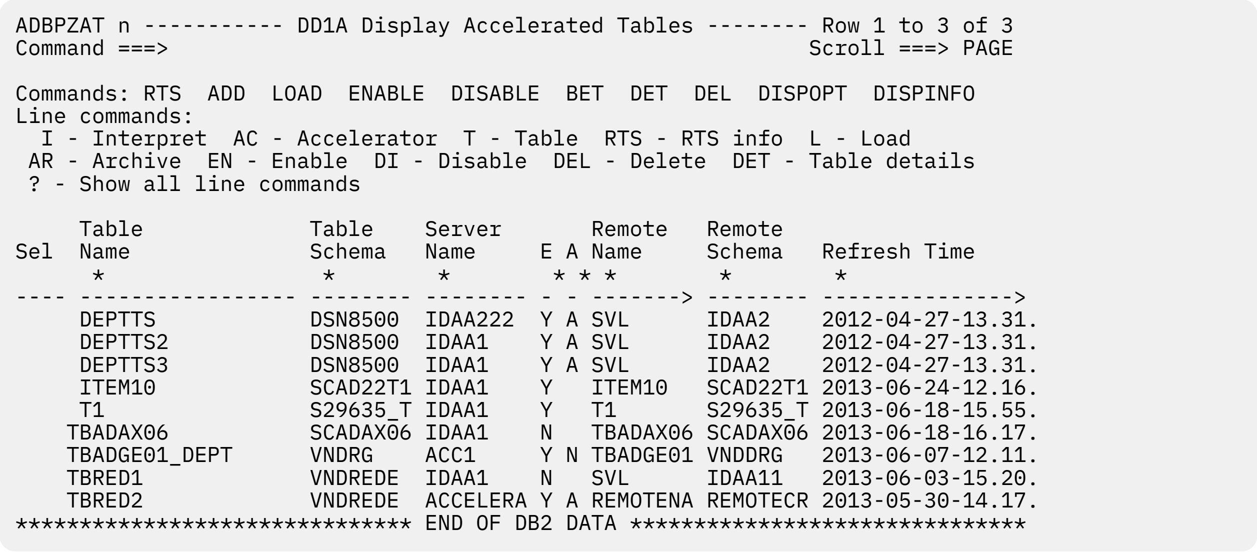
Figure 480. ***Display Accelerated Tables (ADBPZAT)*** *panel*

The status of the accelerated tables is shown in the E (Enable) column. The status is one of the following types:

Y
The table is accelerated.

N
The table is not accelerated.

For example, after you submit the commands in the preceding example, the table ITEM10 is enabled, and the table TBADAX06 is disabled, as shown in the following figure:



```
ADBPZAT n ----------- DD1A Display Accelerated Tables -------- Row 1 to 3 of 3
Command ===>                                                  Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPINFO
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
 ? - Show all line commands

     Table              Table    Server       Remote   Remote
Sel  Name               Schema   Name     E A Name     Schema   Refresh Time
      *                  *        *        * * *        *        *
---- ------------------ -------- -------- - - -------> -------- --------------->
     DEPTTS             DSN8500  IDAA222  Y A SVL      IDAA2    2012-04-27-13.31.
     DEPTTS2            DSN8500  IDAA1    Y A SVL      IDAA2    2012-04-27-13.31.
     DEPTTS3            DSN8500  IDAA1    Y A SVL      IDAA2    2012-04-27-13.31.
     ITEM10             SCAD22T1 IDAA1    Y   ITEM10   SCAD22T1 2013-06-24-12.16.
     T1                 S29635_T IDAA1    Y   T1       S29635_T 2013-06-18-15.55.
    TBADAX06            SCADAX06 IDAA1    N   TBADAX06 SCADAX06 2013-06-18-16.17.
    TBADGE01_DEPT       VNDRG    ACC1     Y N TBADGE01 VNDDRG   2013-06-07-12.11.
    TBRED1              VNDREDE  IDAA1    N   SVL      IDAA11   2013-06-03-15.20.
    TBRED2              VNDREDE  ACCELERA Y A REMOTENA REMOTECR 2013-05-30-14.17.
******************************* END OF DB2 DATA *******************************
```

Figure 481. ***Display Accelerated Tables (ADBPZAT)*** *panel*

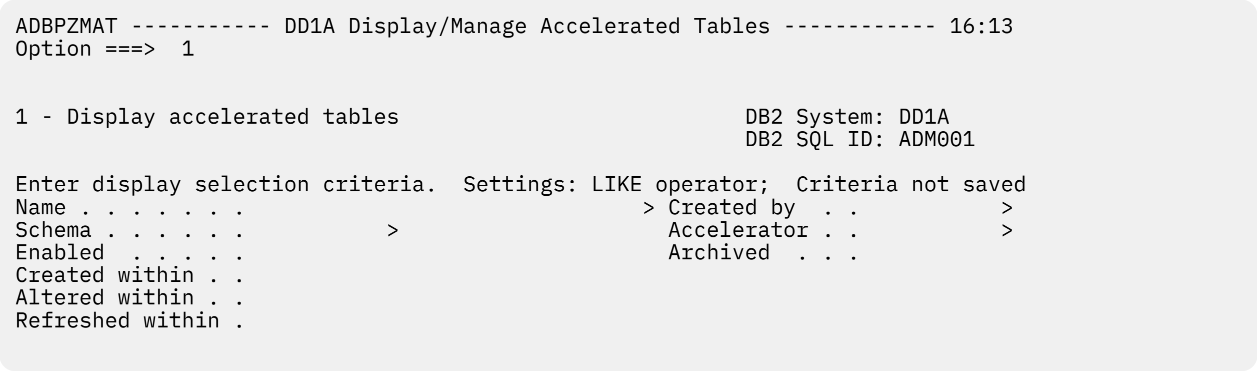
Deleting accelerated tables

You can issue a line or primary command to delete Db2 tables from the accelerator; that is, remove it from the accelerated tables, so that query offloading can be disabled for those tables.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.

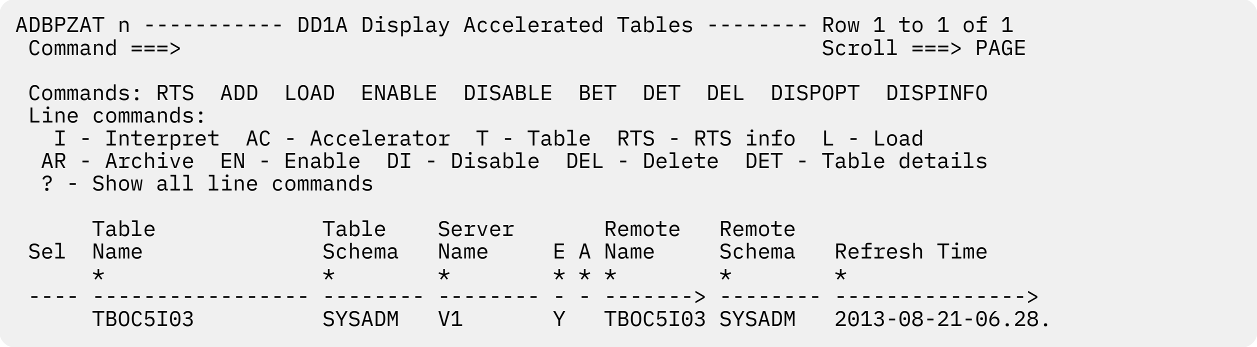
2. On the **System Administration (ADB2Z)** panel, specify option AT, and press Enter.
3. On the **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel, specify any filters for the tables you want to display by entering selection criteria:



```
ADBPZMAT ----------- DD1A Display/Manage Accelerated Tables ------------ 16:13
Option ===>  1


1 - Display accelerated tables                           DB2 System: DD1A
                                                         DB2 SQL ID: ADM001

Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                              > Created by  . .          >
Schema . . . . . .           >                    Accelerator . .          >
Enabled  . . . . .                                Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

Figure 482. ***DB2 Display/Manage Accelerated Tables (ADBPZMAT)*** *panel*

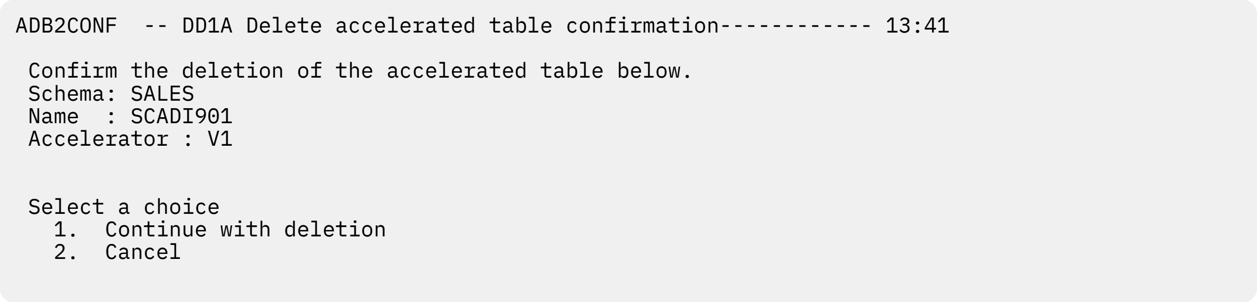
4. Specify option 1, and press Enter.
5. On the **Display Accelerated Tables (ADBPZAT)** panel, specify DEL to delete a table, and press Enter.



```
ADBPZAT n ----------- DD1A Display Accelerated Tables -------- Row 1 to 1 of 1
 Command ===>                                                  Scroll ===> PAGE

 Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPINFO
 Line commands:
   I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
  AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
  ? - Show all line commands

      Table             Table    Server       Remote   Remote
 Sel  Name              Schema   Name     E A Name     Schema   Refresh Time
      *                 *        *        * * *        *        *
 ---- ---------------- -------- -------- - - -------> -------- --------------->
      TBOC5I03          SYSADM   V1       Y   TBOC5I03 SYSADM   2013-08-21-06.28.
```

Figure 483. ***Display Accelerated Tables (ADBPZAT)*** *panel*

6. On the **Delete accelerated table confirmation (ADB2CONF)** panel, specify 1 to proceed.



```
ADB2CONF  -- DD1A Delete accelerated table confirmation------------ 13:41

 Confirm the deletion of the accelerated table below.
 Schema: SALES
 Name  : SCADI901
 Accelerator : V1


 Select a choice
   1.  Continue with deletion
   2.  Cancel
```

Figure 484. Delete accelerated table confirmation panel (ADB2CONF)

Enabling and disabling automatic recreate, reload, or removal of accelerated tables

You can specify whether you want the accelerator to automatically detect changes in Db2 accelerated tables, such as changes to the table definition or dropping data from the Db2 catalog. This feature helps you push Db2 changes to the accelerator automatically and ensures that the table definition and data are synced between Db2 and the accelerator.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option P, and press Enter.

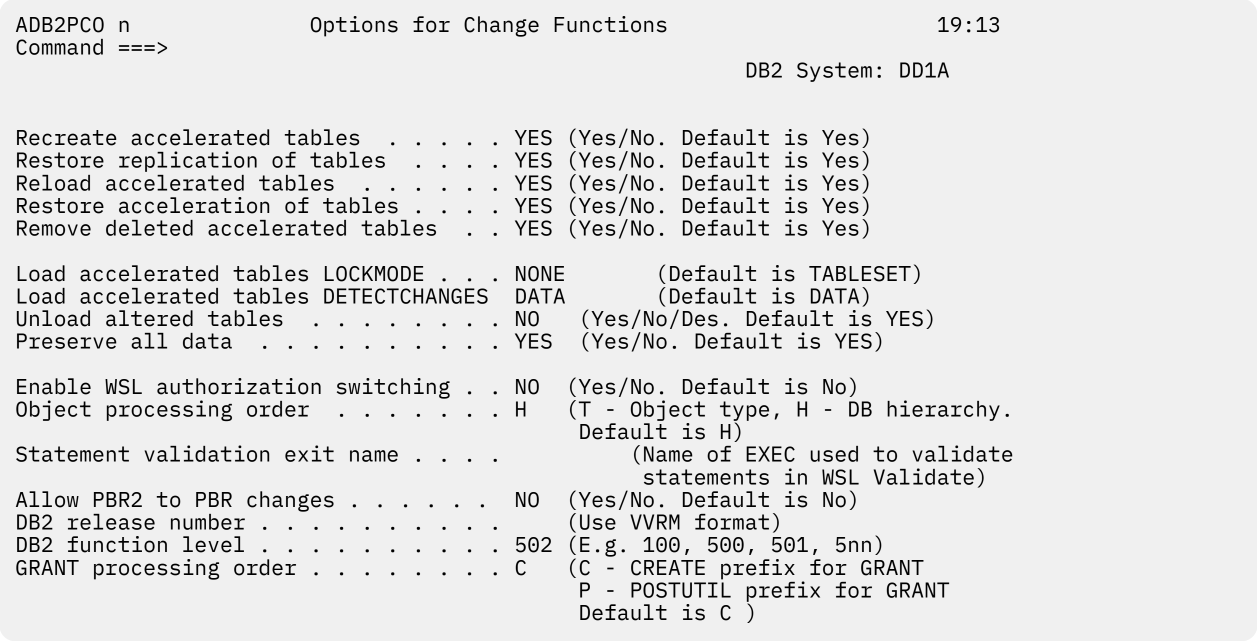
2. On the **DB2 Admin Options (ADB2P)** panel, specify CH, and press Enter.
3. On the **Options for Change Functions (ADB2PCO)** panel, specify YES for the following options on this panel to run the described actions:

   Recreate accelerated tables
   Removes and adds Db2 tables to the accelerators when these tables are dropped and recreated on a Db2 subsystem during an object comparison operation. This feature updates Db2 table definition changes on the accelerator, including distribution or organizing key information that was previously specified for the accelerated tables.

   Reload accelerated tables
   Reloads Db2 data into the accelerated table when a Db2 table is dropped and recreated during an object comparison operation. This option works in conjunction with the **Recreate accelerated tables** option, which updates only the definition changes.

   Remove deleted tables from accelerator
   Removes tables from the accelerator when tables are dropped from Db2 using change management.

   These options operate on existing accelerated tables that are involved in Db2 alter or drop operations.



```
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 LOCKMODE . . . NONE       (Default is TABLESET)
Load accelerated tables DETECTCHANGES  DATA       (Default is DATA)
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)
GRANT processing order . . . . . . . . C   (C - CREATE prefix for GRANT
                                            P - POSTUTIL prefix for GRANT
                                            Default is C )
```

*Figure 485. **Options for Change Functions (ADB2PCO)** panel*

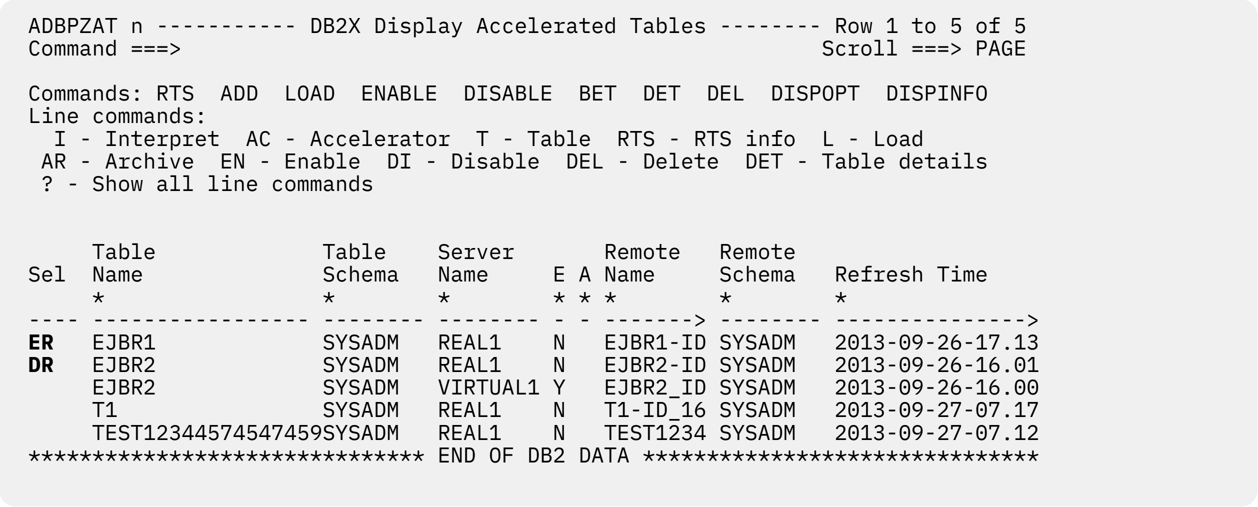
Enabling and disabling incremental updates to accelerated tables

When incremental updates are enabled, updates to tables are automatically propagated to the corresponding tables on the accelerator server with little delay. Disabling incremental updates excludes tables from the incremental update process.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AT, and press Enter.
3. On the **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel, select option 1, and press Enter.
4. On the **Display Accelerated Tables (ADBPZAT)** panel, specify the ER line command to enable incremental updates or the DR line command to disable incremental updates, and press Enter.

   The following example shows both line commands specified:



```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 5 of 5
Command ===>                                                  Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPINFO
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
 ? - Show all line commands


     Table             Table    Server       Remote   Remote
Sel  Name              Schema   Name     E A Name     Schema   Refresh Time
     *                 *        *        * * *        *        *
---- ----------------- -------- -------- - - -------> -------- --------------->
ER   EJBR1             SYSADM   REAL1    N   EJBR1-ID SYSADM   2013-09-26-17.13
DR   EJBR2             SYSADM   REAL1    N   EJBR2-ID SYSADM   2013-09-26-16.01
     EJBR2             SYSADM   VIRTUAL1 Y   EJBR2_ID SYSADM   2013-09-26-16.00
     T1                SYSADM   REAL1    N   T1-ID_16 SYSADM   2013-09-27-07.17
     TEST12344574547459SYSADM   REAL1    N   TEST1234 SYSADM   2013-09-27-07.12
******************************* END OF DB2 DATA *******************************
```

*Figure 486. **Display Accelerated Tables (ADBPZAT)** panel*

Archiving a table partition

You can reduce Db2 storage space by archiving table partitions to IBM Db2 Analytics Accelerator for z/OS (IDAA).

Before you begin

To archive a table, the table must meet the criteria that is listed in Freeing up storage in Db2 for z/OS (IBM DB2 Analytics Accelerator for z/OS 7.5).

About this task

When you archive a table partition to IDAA, Db2 stores active data only and archive data is moved to the accelerator, thus reducing Db2 storage space. Archiving a table partition is valid only when you are using Db2 10 or later.

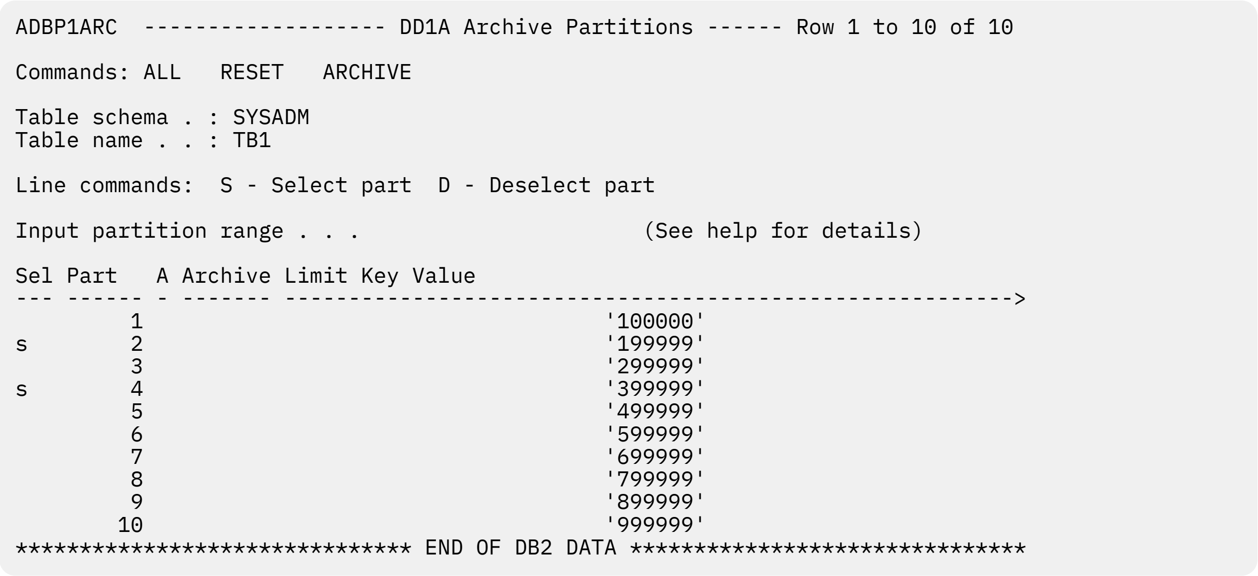
You can select partitions to archive from a selected table on the following panels: ADB21T, ADB21S, ADB21SP, and ADBPZAT. The following procedure uses the ADB21T panel.

Procedure

To archive a table partition:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option T, and press Enter.
3. On the **Tables, Views, and Aliases (ADB21T)** panel, specify the AR line command next to the table to be archived, and press Enter.

   The **Archive Partitions (ADBP1ARC)** panel lists the partitions for the specified table:



```
ADBP1ARC  ------------------- DD1A Archive Partitions ------ Row 1 to 10 of 10

Commands: ALL   RESET   ARCHIVE

Table schema . : SYSADM
Table name . . : TB1

Line commands:  S - Select part  D - Deselect part

Input partition range . . .                      (See help for details)

Sel Part   A Archive Limit Key Value
--- ------ - ------- --------------------------------------------------------->
         1                                    '100000'
s        2                                    '199999'
         3                                    '299999'
s        4                                    '399999'
         5                                    '499999'
         6                                    '599999'
         7                                    '699999'
         8                                    '799999'
         9                                    '899999'
        10                                    '999999'
******************************* END OF DB2 DATA *******************************
```

Figure 487. ***Archive Partitions (ADBP1ARC)*** *panel*

4. Specify the partitions that you want to archive by taking one or more of the following actions:

   - Use the S line command to select individual partitions.
   - Use the D line command to deselect a partition.
   - Specify ALL to select all partitions.
   - Specify RESET to clear all selections.
   - Specify a partition range in the **Input partition range** field. Input the partitions to archive by using the same syntax as the SYSPROC.ACCEL_ARCHIVE_TABLES stored procedure. Use a colon (:) to specify a range. Use a negative number to specify partitions that start from the last partition. For example, -2 specifies the second-to-last partition. The following examples show valid partition range values:

     1, 2
     Specifies partitions 1 and 2.

     1, 2:3
     Specifies partitions 1, 2, and 3.

     1:2,3
     Specifies partitions 1, 2, and 3.

     -2
     Selects the second to last partition.

     -2:-1
     Selects the second to last partition and the last partition.

     -3,-1
     Selects the third to last partition and the last partition.

   After you press Enter, the tables that are marked for archiving display Y in the **Archive** column.

   Important: You can archive table partitions in batch to reduce wait time. You can run batch by specifying YES for **Run Accelerator functions in batch** on the **DB2 Admin Defaults (ADB2P2)** panel.

5. After all of the partitions have been specified, issue the ARCHIVE primary command, and press Enter.

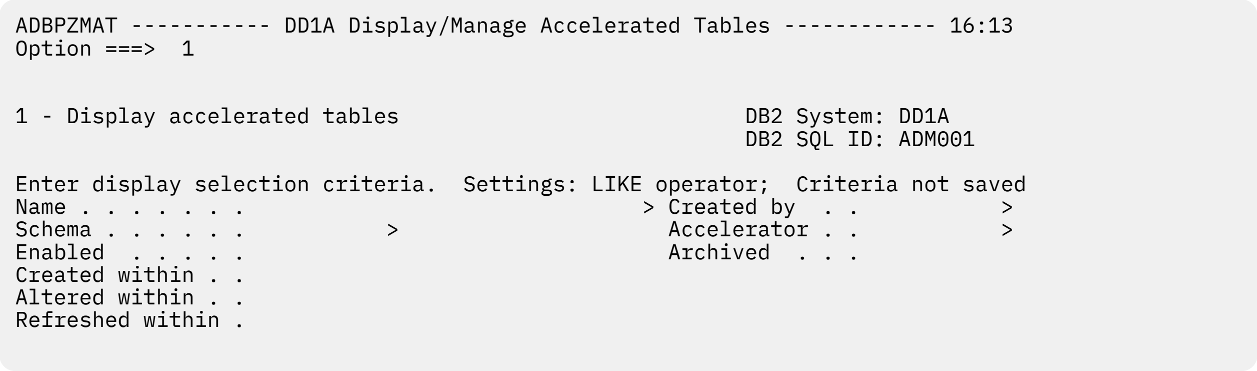
   Db2 Admin Tool calls the stored procedure that archives the partitions to the accelerated table database.

Restoring archived accelerated tables

You can restore an archived table in Db2.

Procedure

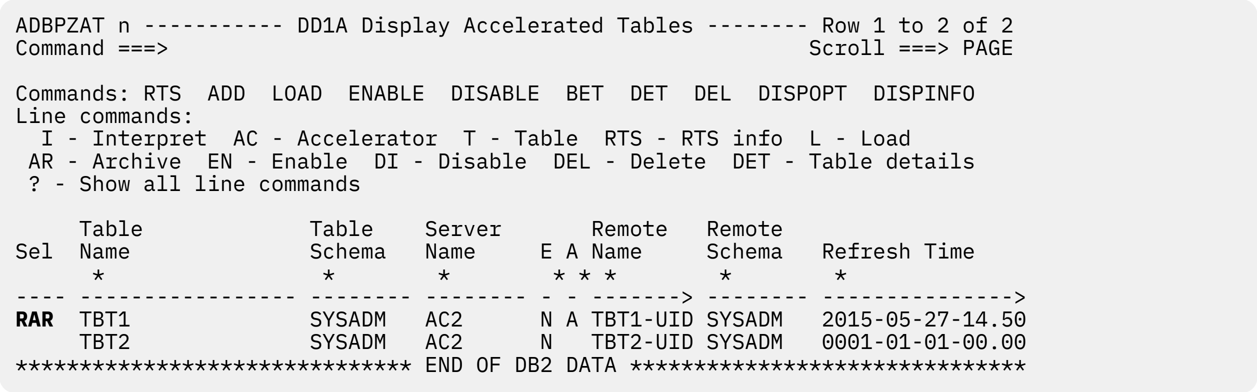
1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AT, and press Enter.
3. On the **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel, specify any filters for the tables you want to display by entering selection criteria:



```
ADBPZMAT ----------- DD1A Display/Manage Accelerated Tables ----------- 16:13
Option ===>  1


1 - Display accelerated tables                          DB2 System: DD1A
                                                        DB2 SQL ID: ADM001

Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                              > Created by  . .           >
Schema . . . . . .           >                      Accelerator . .           >
Enabled  . . . . .                                  Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

*Figure 488. **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel*

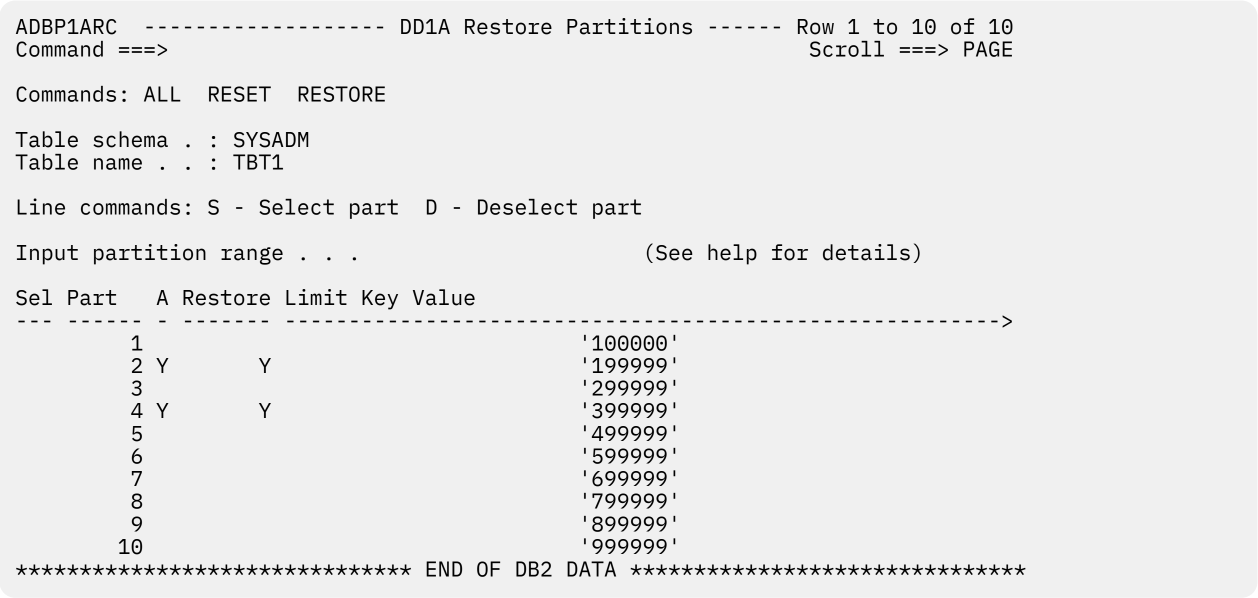
4. Specify option 1, and press Enter.
5. On the **Display Accelerated Tables (ADBPZAT)** panel, specify the RAR line command to restore an archived table.



```
ADBPZAT n ----------- DD1A Display Accelerated Tables -------- Row 1 to 2 of 2
Command ===>                                                 Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPINFO
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
 ? - Show all line commands

     Table             Table    Server       Remote   Remote
Sel  Name              Schema   Name     E A Name     Schema   Refresh Time
      *                 *        *        * * *        *        *
---- ----------------- -------- -------- - - -------> -------- --------------->
RAR  TBT1              SYSADM   AC2      N A TBT1-UID SYSADM   2015-05-27-14.50
     TBT2              SYSADM   AC2      N   TBT2-UID SYSADM   0001-01-01-00.00
******************************* END OF DB2 DATA *******************************
```

*Figure 489. **Display Accelerated Tables (ADBPZAT)** panel*

6. On the **Restore Partitions (ADBP1ARC)** panel, specify the S line command to select the archived partitions that you want to restore.



```
ADBP1ARC  ------------------- DD1A Restore Partitions ------ Row 1 to 10 of 10
Command ===>                                                  Scroll ===> PAGE

Commands: ALL  RESET  RESTORE

Table schema . : SYSADM
Table name . . : TBT1

Line commands: S - Select part  D - Deselect part

Input partition range . . .                      (See help for details)

Sel Part   A Restore Limit Key Value
--- ------ - ------- -------------------------------------------------------->
         1                                  '100000'
         2 Y       Y                        '199999'
         3                                  '299999'
         4 Y       Y                        '399999'
         5                                  '499999'
         6                                  '599999'
         7                                  '699999'
         8                                  '799999'
         9                                  '899999'
        10                                  '999999'
******************************* END OF DB2 DATA *******************************
```

Figure 490. ***Restore Partitions (ADBP1ARC)*** *panel*

In column A, a Y value indicates that the table is archived. In the Restore column, a Y value indicates that the table will be restored.

7. After you have selected the archived partitions, issue the RESTORE primary command.

Viewing real-time status information for accelerated tables

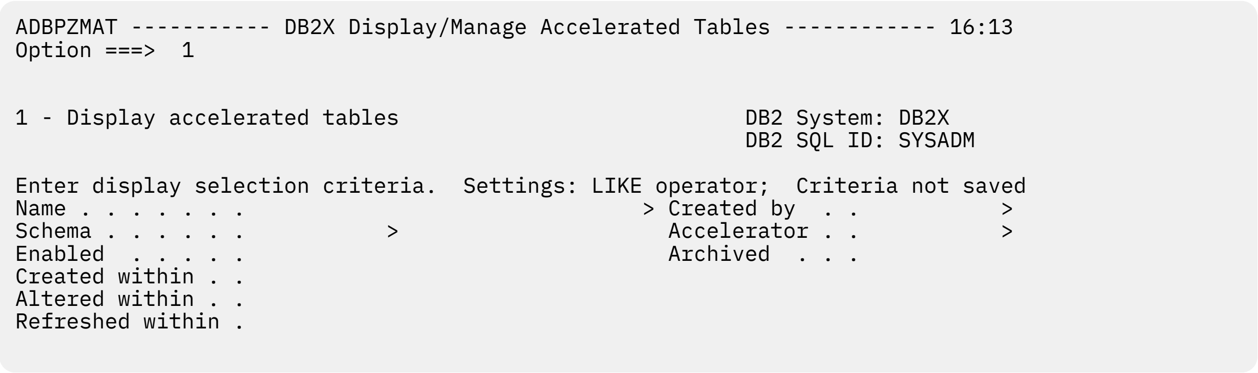
You can view RUNSTATS and real-time status information for accelerated tables to help you decide whether to reload the accelerated table.

About this task

Information about the accelerated tables is stored in the pseudo-catalog table, SYSACCEL.SYSACCELERATEDTABLES. Each Db2 connection instance has one pseudo-catalog table.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AT, and press Enter.
3. On the **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel, specify any filters for the tables you want to display by entering selection criteria:

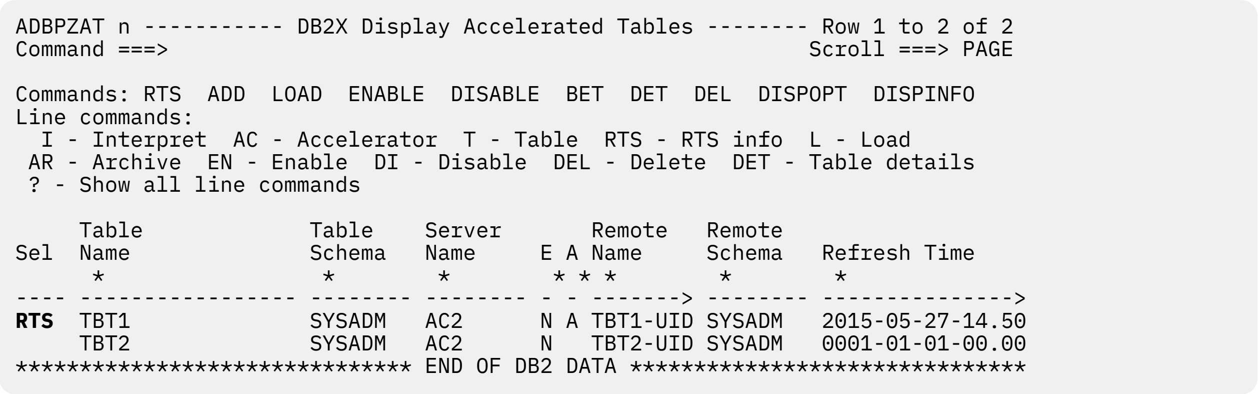


```
ADBPZMAT ----------- DB2X Display/Manage Accelerated Tables ----------- 16:13
Option ===>  1


1 - Display accelerated tables                           DB2 System: DB2X
                                                         DB2 SQL ID: SYSADM

Enter display selection criteria.  Settings: LIKE operator;  Criteria not saved
Name . . . . . . .                                > Created by  . .           >
Schema . . . . . .           >                      Accelerator . .           >
Enabled  . . . . .                                  Archived  . . .
Created within . .
Altered within . .
Refreshed within .
```

Figure 491. ***DB2 Display/Manage Accelerated Tables (ADBPZMAT)*** *panel*

4. Specify option 1, and press Enter.

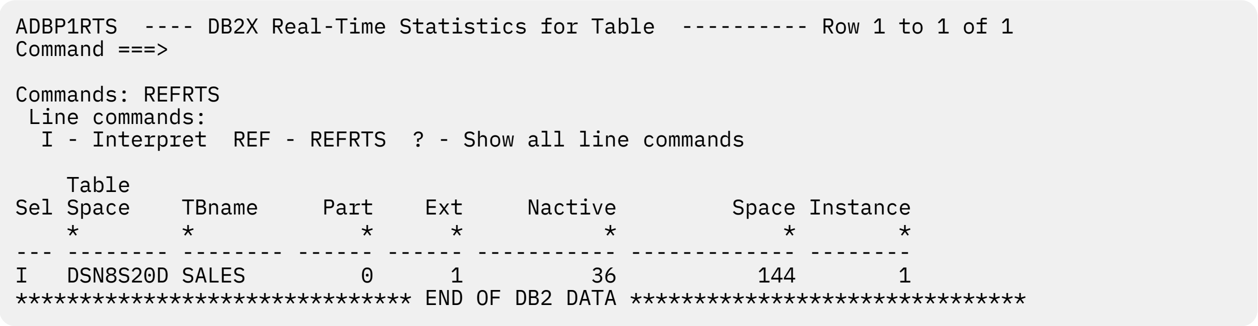
5. On the **Display Accelerated Tables (ADBPZAT)** panel, specify the RTS line command to display real-time statistics for a particular table, and press Enter. Alternatively, you can issue the RTS primary command to display real-time statistics for all the tables on the panel.



```
ADBPZAT n ----------- DB2X Display Accelerated Tables -------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPINFO
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
 ? - Show all line commands

     Table             Table    Server       Remote   Remote
Sel  Name              Schema   Name     E A Name     Schema   Refresh Time
      *                 *        *        * * *        *        *
---- ----------------- -------- -------- - - -------> -------- --------------->
RTS  TBT1              SYSADM   AC2      N A TBT1-UID SYSADM   2015-05-27-14.50
     TBT2              SYSADM   AC2      N   TBT2-UID SYSADM   0001-01-01-00.00
******************************* END OF DB2 DATA *******************************
```

*Figure 492. **Display Accelerated Tables (ADBPZAT)** panel*

The following example shows the results of issuing the RTS line command:

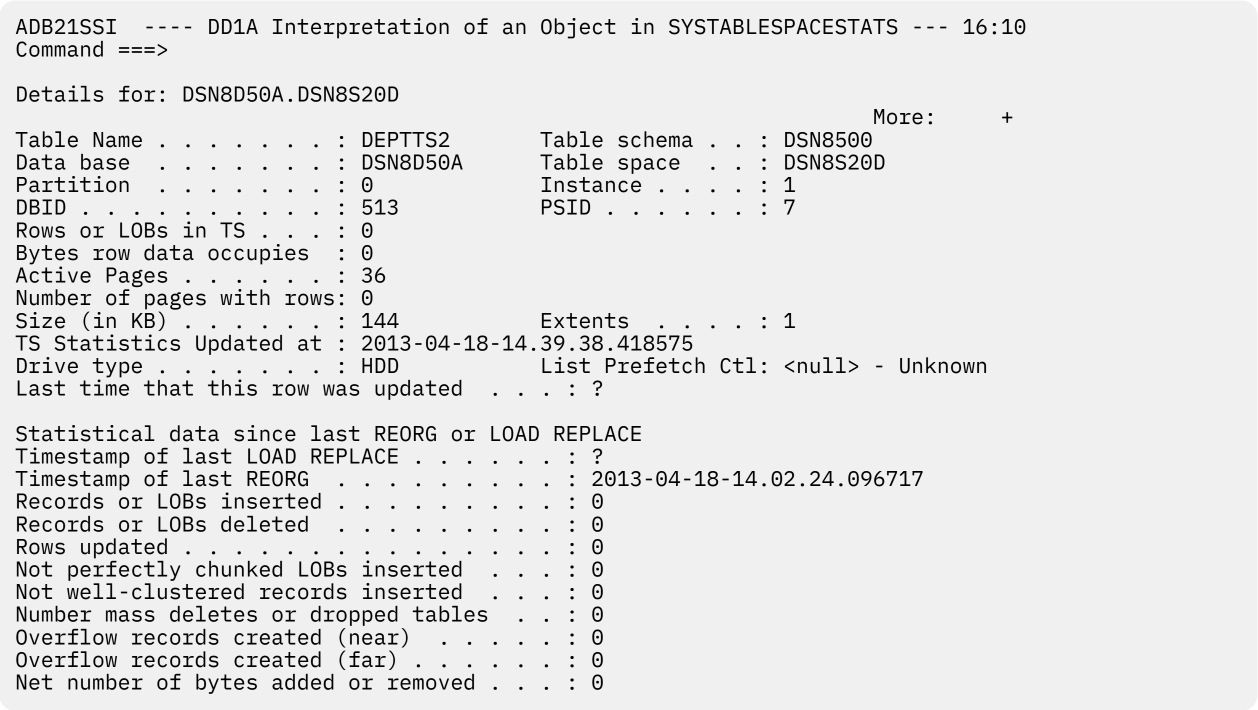


```
ADBP1RTS  ---- DB2X Real-Time Statistics for Table  ---------- Row 1 to 1 of 1
Command ===>

Commands: REFRTS
 Line commands:
  I - Interpret  REF - REFRTS  ? - Show all line commands

    Table
Sel Space    TBname     Part    Ext     Nactive         Space Instance
    *        *             *      *           *             *        *
--- -------- -------- ------ ------ ----------- ------------- --------
I   DSN8S20D SALES         0      1          36           144        1
******************************* END OF DB2 DATA *******************************
```

*Figure 493. **Table Space Maintenance (ADB2314)** panel*

6. Specify the I line command to display more detailed RUNSTATS information for the table, and press Enter.

   The **Interpretation of an Object in SYSTABLESPACESTATS (ADB21SSI)** panel is displayed, as shown in the following figure:



```
ADB21SSI  ---- DD1A Interpretation of an Object in SYSTABLESPACESTATS --- 16:10
Command ===>

Details for: DSN8D50A.DSN8S20D
                                                                   More:     +
Table Name . . . . . . . : DEPTTS2       Table schema . . : DSN8500
Data base  . . . . . . . : DSN8D50A      Table space  . . : DSN8S20D
Partition  . . . . . . . : 0             Instance . . . . : 1
DBID . . . . . . . . . . : 513           PSID . . . . . . : 7
Rows or LOBs in TS . . . : 0
Bytes row data occupies  : 0
Active Pages . . . . . . : 36
Number of pages with rows: 0
Size (in KB) . . . . . . : 144           Extents  . . . . : 1
TS Statistics Updated at : 2013-04-18-14.39.38.418575
Drive type . . . . . . . : HDD           List Prefetch Ctl: <null> - Unknown
Last time that this row was updated  . . . : ?

Statistical data since last REORG or LOAD REPLACE
Timestamp of last LOAD REPLACE . . . . . . : ?
Timestamp of last REORG  . . . . . . . . . : 2013-04-18-14.02.24.096717
Records or LOBs inserted . . . . . . . . . : 0
Records or LOBs deleted  . . . . . . . . . : 0
Rows updated . . . . . . . . . . . . . . . : 0
Not perfectly chunked LOBs inserted  . . . : 0
Not well-clustered records inserted  . . . : 0
Number mass deletes or dropped tables  . . : 0
Overflow records created (near)  . . . . . : 0
Overflow records created (far) . . . . . . : 0
Net number of bytes added or removed . . . : 0
```

Figure 494. ***Interpretation of an Object in SYSTABLESPACESTATS (ADB21SSI)*** *panel*

The table name and table schema are displayed together with database and table space information. RUNSTATS information is based on the table space.

Viewing accelerated table details

You can create reports that show details for each accelerated table, including change and archive information for the entire table or in the case of partitioned tables, for each partition.

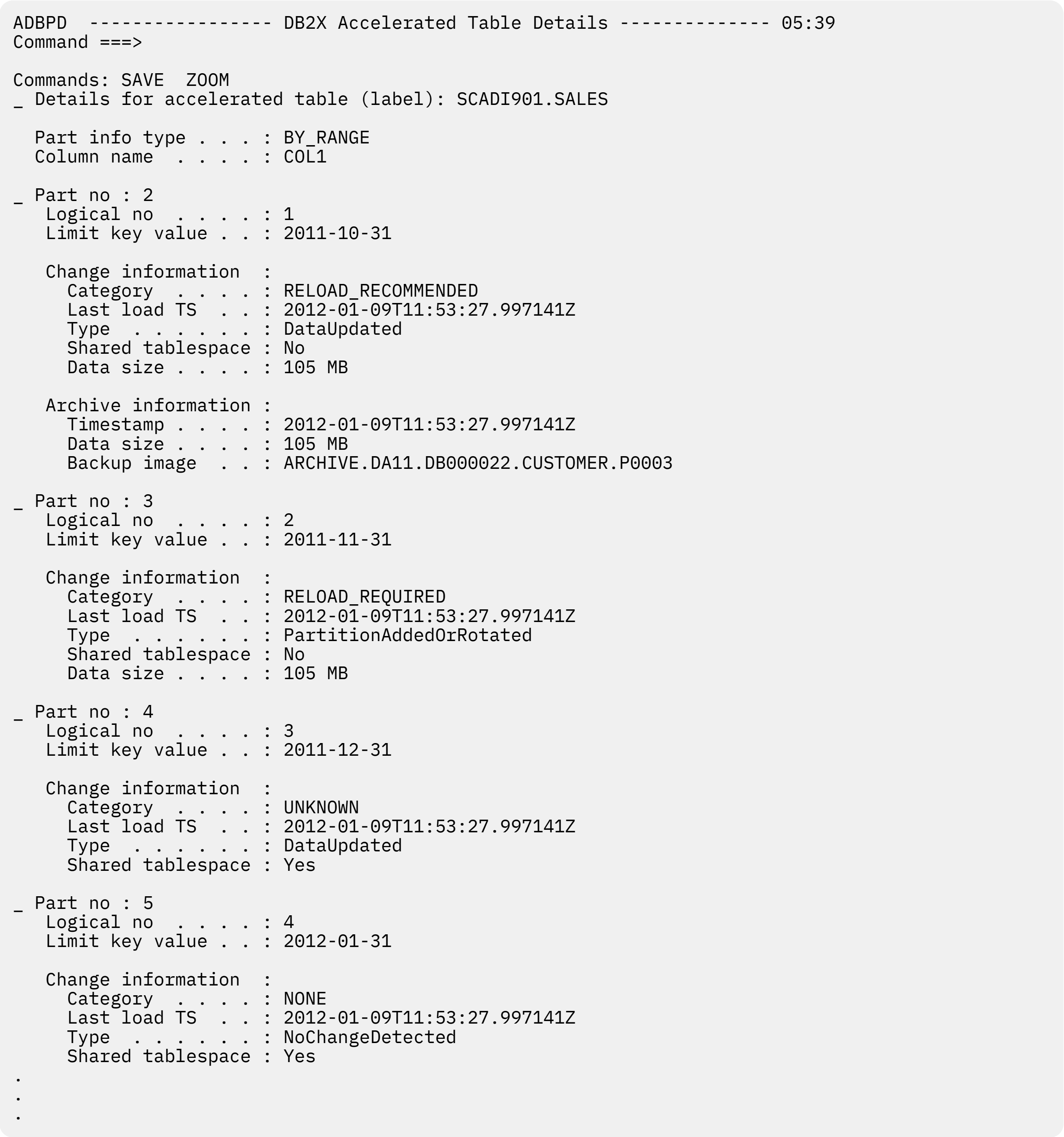
About this task

As an alternative to this procedure, you can view accelerated table information by using the commands on **Display Accelerated Tables (ADBPZAT)** panel. See “Displaying accelerator tables” on page 948.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AC, and press Enter.
3. On the **DB2 Accelerators (ADBPZAC)** panel, specify the DET command, and press Enter.

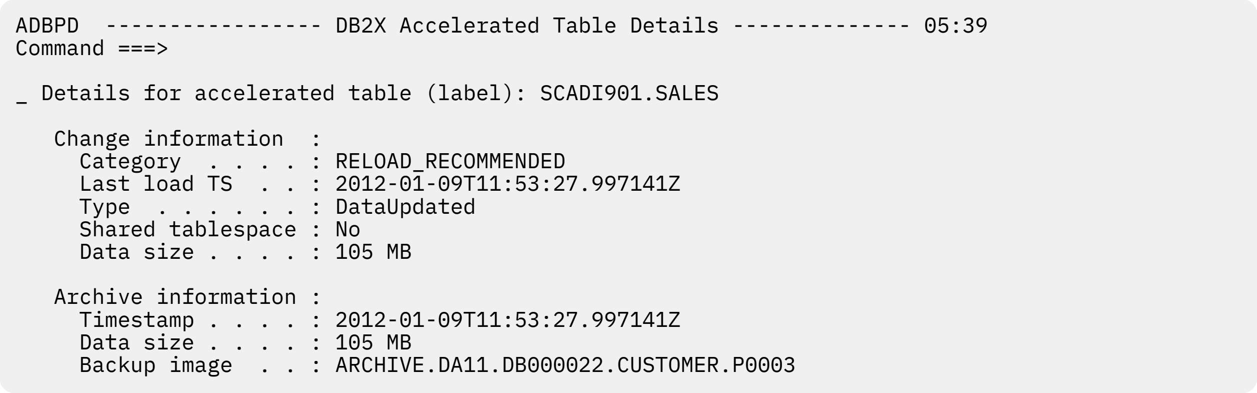
   The **Accelerated Table Details (ADBPD)** panel is displayed. The following example panel is for a partitioned table.



```
ADBPD  ----------------- DB2X Accelerated Table Details -------------- 05:39
Command ===>

Commands: SAVE  ZOOM
_ Details for accelerated table (label): SCADI901.SALES

  Part info type . . . : BY_RANGE
  Column name  . . . . : COL1

_ Part no : 2
   Logical no  . . . . : 1
   Limit key value . . : 2011-10-31

   Change information  :
     Category  . . . . : RELOAD_RECOMMENDED
     Last load TS  . . : 2012-01-09T11:53:27.997141Z
     Type  . . . . . . : DataUpdated
     Shared tablespace : No
     Data size . . . . : 105 MB

   Archive information :
     Timestamp . . . . : 2012-01-09T11:53:27.997141Z
     Data size . . . . : 105 MB
     Backup image  . . : ARCHIVE.DA11.DB000022.CUSTOMER.P0003

_ Part no : 3
   Logical no  . . . . : 2
   Limit key value . . : 2011-11-31

   Change information  :
     Category  . . . . : RELOAD_REQUIRED
     Last load TS  . . : 2012-01-09T11:53:27.997141Z
     Type  . . . . . . : PartitionAddedOrRotated
     Shared tablespace : No
     Data size . . . . : 105 MB

_ Part no : 4
   Logical no  . . . . : 3
   Limit key value . . : 2011-12-31

   Change information  :
     Category  . . . . : UNKNOWN
     Last load TS  . . : 2012-01-09T11:53:27.997141Z
     Type  . . . . . . : DataUpdated
     Shared tablespace : Yes

_ Part no : 5
   Logical no  . . . . : 4
   Limit key value . . : 2012-01-31

   Change information  :
     Category  . . . . : NONE
     Last load TS  . . : 2012-01-09T11:53:27.997141Z
     Type  . . . . . . : NoChangeDetected
     Shared tablespace : Yes
.
.
.
```

Figure 495. ***Accelerated Table Details (ADBPD)*** *panel*

For non-partitioned tables, information is displayed for the entire table instead of the individual partitions, as shown in the following figure:



```
ADBPD  ----------------- DB2X Accelerated Table Details -------------- 05:39
Command ===>

_ Details for accelerated table (label): SCADI901.SALES

   Change information  :
     Category  . . . . : RELOAD_RECOMMENDED
     Last load TS  . . : 2012-01-09T11:53:27.997141Z
     Type  . . . . . . : DataUpdated
     Shared tablespace : No
     Data size . . . . : 105 MB

   Archive information :
     Timestamp . . . . : 2012-01-09T11:53:27.997141Z
     Data size . . . . : 105 MB
     Backup image  . . : ARCHIVE.DA11.DB000022.CUSTOMER.P0003
```

*Figure 496. **Accelerated Table Details (ADBPD)** panel*

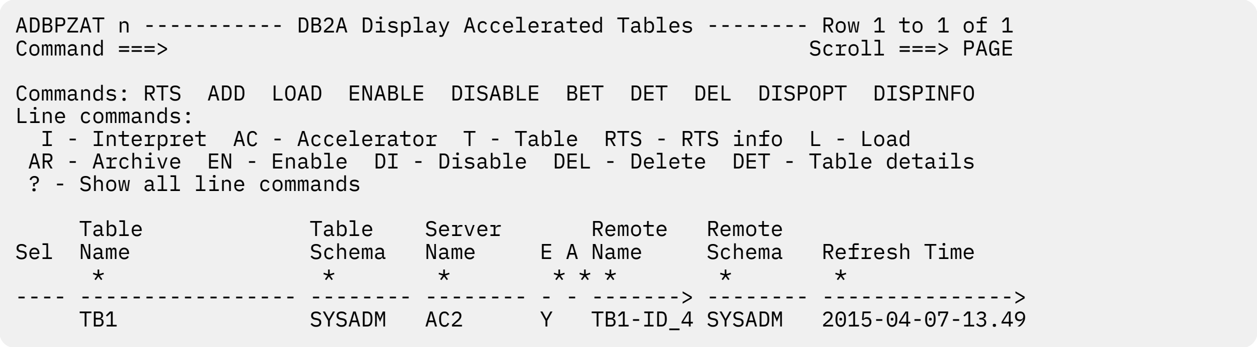
Specifying or altering distribution and organizing keys

You can specify distribution or organizing keys for accelerated tables. Distribution keys distribute data across multiple nodes, and organizing keys sort table rows into blocks. Applying these keys to your tables can improve query response times.

Procedure

To specify or alter distribution and organizing keys:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option AT, and press Enter.
3. Optional: On the **DB2 Display/Manage Accelerated Tables (ADBPZMAT)** panel. specify selection criteria to filter the tables that you want to display.
4. Specify option 1, and press Enter.
5. On the **Display Accelerated Tables (ADBPZAT)** panel, specify KEYS next to the name of the table for which you want to specify a distribution or organizing key, and press Enter.



```
ADBPZAT n ----------- DB2A Display Accelerated Tables -------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Commands: RTS  ADD  LOAD  ENABLE  DISABLE  BET  DET  DEL  DISPOPT  DISPINFO
Line commands:
  I - Interpret  AC - Accelerator  T - Table  RTS - RTS info  L - Load
 AR - Archive  EN - Enable  DI - Disable  DEL - Delete  DET - Table details
 ? - Show all line commands

     Table             Table    Server       Remote   Remote
Sel  Name              Schema   Name     E A Name     Schema   Refresh Time
      *                 *        *        * * *        *        *
---- ----------------- -------- -------- - - -------> -------- --------------->
     TB1               SYSADM   AC2      Y   TB1-ID_4 SYSADM   2015-04-07-13.49
```

6. On the **Accelerated Table - Keys (ADBPZAK)** panel, specify a line command to indicate the relative position of the column in the distribution or organizing key.

   You can specify any of the following valid values for distribution keys and organizing keys:

   Distribution keys
   D1, D2, D3, and D4.

   Organizing keys
   O1, O2, O3, and O4.

```
ADBPZAK   ------ DC1Q Accelerated Table - Keys --------------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> CSR

Commands: SAVE  CANCEL

 Status for table . . : SYSADM.T1
 Accelerator name . . :
IDAAZ12
 Acceleration status  : true       Load status : Loaded
 Replication status . : false      Details . . :

 Used disk space  . . : 1       (in MB)   Row count . . . . . : 16
 Skew . . . . . . . . : 1.000   (0 to 1)  Organized percent . : 0.000

Line commands:  Dn - Distribution sequence  On - Organizing sequence
                 R - Remove column sequence

S  Column Name        D O Col Type      Length  Scale Nulls
   *                  * * *                  *      * *
-- ------------------ - - -------- ----------- ------ ------
   COL1                   INTEGER            4      0 N
   COL2                   CHAR              10      0 Y
******************************* END OF DB2 DATA *******************************
```

7. Issue the SAVE command to save your changes.

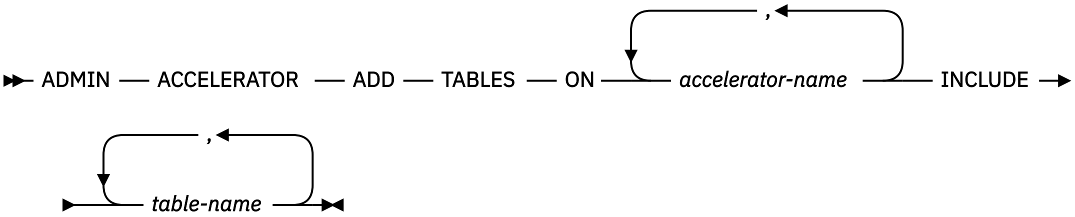
ADMIN ACCELERATOR statements

Db2 Admin Tool uses ADMIN ACCELERATOR statements to perform various accelerator tasks on IBM Db2 Analytics Accelerator for z/OS.

ADMIN ACCELERATOR ADD statement

The ADMIN ACCELERATOR ADD statement defines a Db2 table on an accelerator.

Syntax



Option descriptions

accelerator-name

The name of the accelerator on which you want to define the specified table or tables.

You can specify more than one accelerator. Separate each name with a comma.

INCLUDE *table-name*

Specifies the table or tables that are to be defined on the accelerator. If multiple accelerators are specified in the statement, the table or tables are added to each accelerator.

table-name

The qualified name of the table (*schema.name*).

If you specify more than one table name, separate each name by a comma.

If you use wildcard characters, you can specify only one table name, and you must enclose the name or schema with a wildcard character in quotation marks. You can use the following wildcard characters:

_

Indicates a single character

%
: Indicates a varying length string

*****
: Indicates a varying length string

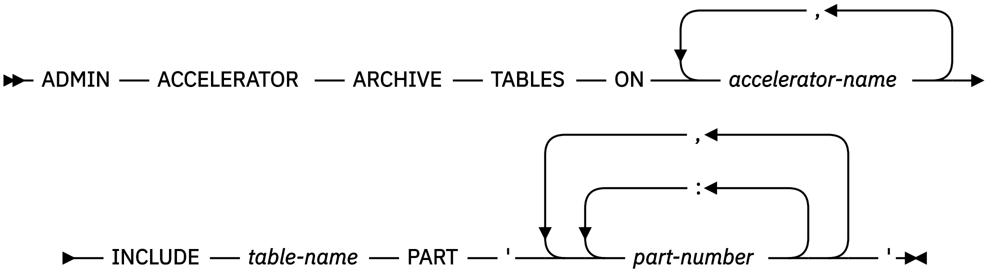
Example table name specifications:

```
ADMF001.TBT1,ADMF002.TBT2
"ADMF%"."TBT_"
SYSADM."*"
```

ADMIN ACCELERATOR ARCHIVE statement

The ADMIN ACCELERATOR ARCHIVE statement archives table partitions in partition-by-range table spaces on one or more accelerators.

Syntax



Option descriptions

accelerator-name
The name of the accelerator on which you want to archive the specified partitions.

You can specify more than one accelerator. Separate each name with a comma.

INCLUDE *table-name* PART '*part-number*'
Specifies the table partitions that are to be archived on the accelerator. If multiple accelerators are specified in the statement, the partitions are archived on each accelerator.

table-name
The qualified name of the table (*schema.name*).

You can use the following wildcard characters in the table name:

_
Indicates a single character

%
Indicates a varying length string

*****
Indicates a varying length string

If you use wildcard characters, enclose the name or schema with a wildcard character in quotation marks. For example:

```
SYSADM."*"
```

part-number
The logical partition number. You can specify more than one partition. To specify a range, use a colon (:). To separate individual partition numbers and ranges, use a comma (,).

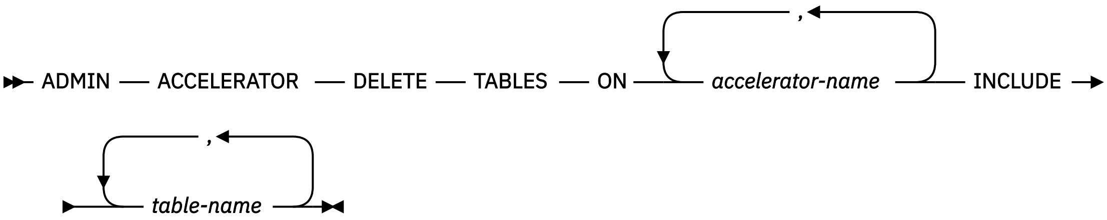
The following table shows examples of PART specifications:

PART specification	Meaning
`PART '1'`	Include the first partition only
`PART '1:10'`	Include the first 10 partitions
`PART '1, 10:13'`	Include the first partition and partitions 10 through 13
`PART '1:3, 5:8'`	Include partitions 1 through 3 and 5 through 8

ADMIN ACCELERATOR DELETE statement

The ADMIN ACCELERATOR DELETE statement deletes a table from the accelerator.

Syntax



Option descriptions

accelerator-name

The name of the accelerator from which you want to delete the specified table or tables.

You can specify more than one accelerator. Separate each name with a comma.

INCLUDE *table-name*

Specifies the table or tables that are to be deleted from the accelerator. If multiple accelerators are specified in the statement, the table or tables are deleted from each accelerator.

table-name

The qualified name of the table (*schema.name*).

If you specify more than one table name, separate each name by a comma.

If you use wildcard characters, you can specify only one table name, and you must enclose the name or schema with a wildcard character in quotation marks. You can use the following wildcard characters:

_

Indicates a single character

%

Indicates a varying length string

*****

Indicates a varying length string

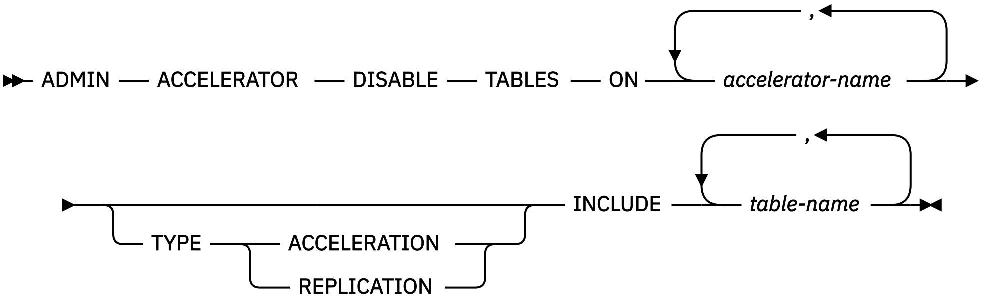
Example table name specifications:

```
ADMF001.TBT1,ADMF002.TBT2
"ADMF%"."TBT_"
SYSADM."*"
```

ADMIN ACCELERATOR DISABLE statement

The ADMIN ACCELERATOR DISABLE statement disables acceleration or replication for a table or tables that are defined on one or more accelerators.

Syntax



Option descriptions

accelerator-name
The name of the accelerator on which you want to disable acceleration or replication for the specified table or tables.

You can specify more than one accelerator. Separate each name with a comma.

TYPE
Specifies the process to be disabled.

ACCELERATION
Acceleration is to be disabled.

REPLICATION
Replication is to be disabled.

If TYPE is not specified, the default value, TYPE ACCELERATION, is used.

INCLUDE *table-name*
Specifies the table or tables for which acceleration or replication is to be disabled. If multiple accelerators are specified in the statement, acceleration or replication is disabled for the specified tables on each accelerator.

table-name
The qualified name of the table (*schema.name*).

If you specify more than one table name, separate each name by a comma.

If you use wildcard characters, you can specify only one table name, and you must enclose the name or schema with a wildcard character in quotation marks. You can use the following wildcard characters:

_
Indicates a single character

%
Indicates a varying length string

*****
Indicates a varying length string

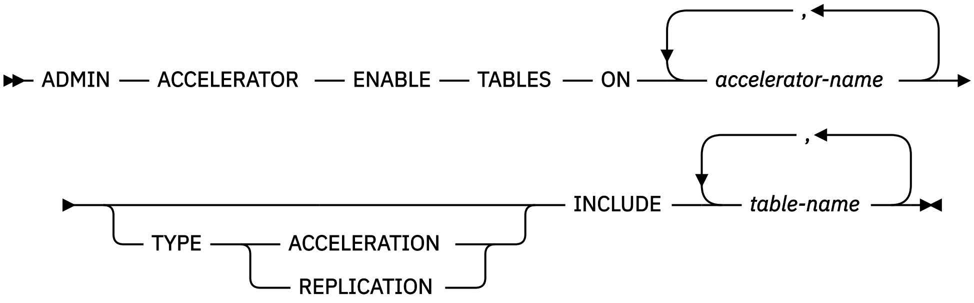
Example table name specifications:

```
ADMF001.TBT1,ADMF002.TBT2
"ADMF%"."TBT_"
SYSADM."*"
```

ADMIN ACCELERATOR ENABLE statement

The ADMIN ACCELERATOR ENABLE statement enables acceleration or replication for a table or tables that are defined on one or more accelerators.

Syntax



Option descriptions

accelerator-name
The name of the accelerator on which you want to enable acceleration or replication for the specified table or tables.

You can specify more than one accelerator. Separate each name with a comma.

TYPE
Specifies the process to be enabled.

ACCELERATION
Acceleration is to be enabled.

REPLICATION
Replication is to be enabled.

If TYPE is not specified, the default value, TYPE ACCELERATION, is used.

INCLUDE *table-name*
Specifies the table or tables for which acceleration or replication is to be enabled. If multiple accelerators are specified in the statement, acceleration or replication is enabled for the specified tables on each accelerator.

table-name
The qualified name of the table (*schema.name*).

If you specify more than one table name, separate each name by a comma.

If you use wildcard characters, you can specify only one table name, and you must enclose the name or schema with a wildcard character in quotation marks. You can use the following wildcard characters:

_
Indicates a single character

%
Indicates a varying length string

*****
Indicates a varying length string

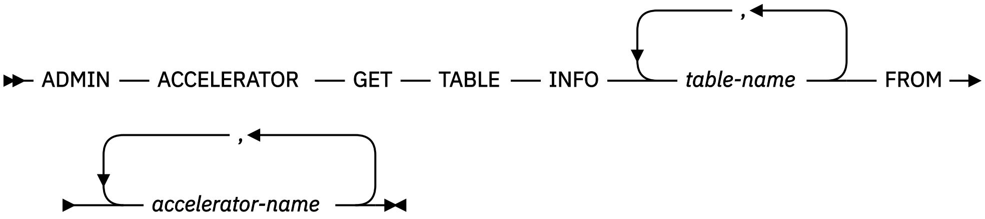
Example table name specifications:

```
ADMF001.TBT1,ADMF002.TBT2
"ADMF%"."TBT_"
SYSADM."*"
```

ADMIN ACCELERATOR GET TABLE INFO statement

The ADMIN ACCELERATOR GET TABLE INFO statement retrieves accelerated table information that is used to restore its acceleration and replication status.

Syntax



Option descriptions

table-name
The qualified name of the table (*schema.name*).

If you specify more than one table name, separate each name by a comma.

If you use wildcard characters, you can specify only one table name, and you must enclose the name or schema with a wildcard character in quotation marks. You can use the following wildcard characters:

_
Indicates a single character

%
Indicates a varying length string

*****
Indicates a varying length string

Example table name specifications:

```
ADMF001.TBT1,ADMF002.TBT2
"ADMF%"."TBT_"
SYSADM."*"
```

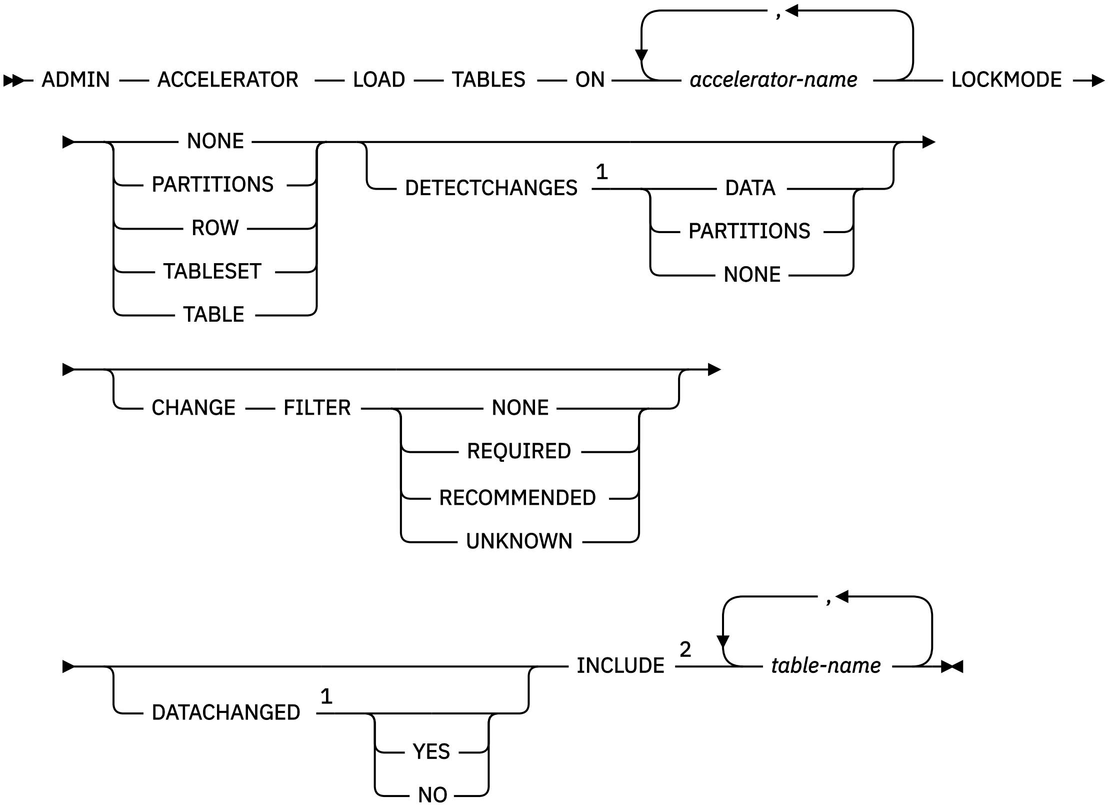
accelerator-name
The name of the accelerator on which the table is defined.

You can specify more than one accelerator. Separate each name with a comma.

ADMIN ACCELERATOR LOAD statement

The ADMIN ACCELERATOR LOAD statement loads Db2 table data from accelerated tables into the accelerator.

Syntax



Notes:

[1] DETECTCHANGES and DATACHANGED are mutually exclusive. Although DATACHANGED is supported, it is deprecated; DETECTCHANGES is the preferred option.

[2] If CHANGE FILTER or DATACHANGED is specified, the INCLUDE clause is optional.

Option descriptions

accelerator-name

The name of the accelerator into which you want to load data.

You can specify more than one accelerator. Separate each name with a comma.

LOCKMODE

Controls the protection level while accelerator-shadow tables are being loaded. Possible values are: NONE, PARTITIONS, ROW, TABLESET, and TABLE. The default is TABLESET.

For a description of each of these values, see the description of *lock_mode* in SYSPROC.ACCEL_LOAD_TABLES (IBM DB2 Analytics Accelerator for z/OS 7.5).

DETECTCHANGES

Specifies the value of detectChanges for the SYSPROC.ACCEL_LOAD_TABLES procedure for accelerated tables. Possible values are: DATA, PARTITIONS, or NONE. DATA is the default value.

For information about DATA and PARTITIONS, see the description of *detectChanges* in SYSPROC.ACCEL_LOAD_TABLES (IBM DB2 Analytics Accelerator for z/OS 7.5).

NONE indicates that detectChanges is not used when calling SYSPROC.ACCEL_LOAD_TABLES.

CHANGE FILTER
Restricts the accelerated tables to which the ADMIN ACCELERATOR LOAD statement is applied. Possible values are:

NONE
Specifies that the statement applies to only tables without changed data.

REQUIRED
Specifies that the statement applies to only tables that require the table or partition to be reloaded. This situation occurs when partitioning of the table is not in sync with the currently loaded table on the accelerator.

RECOMMENDED
Specifies that the statement applies to only tables whose data has changed but reloading the data is not required because the data differences are tolerable.

UNKNOWN
Specifies that the statement applies to only tables whose available change information is insufficient, inconsistent, or cannot be determined.

By default, the CHANGE FILTER clause is omitted, which means that all tables are processed.

DATACHANGED
Specifies whether to restrict the loading of tables to only those tables where data has changed since the table was last loaded into the accelerator.

YES
Only those tables with data changes are loaded.

NO
Only tables without data changes are loaded.

By default, DATACHANGED is omitted, which means that all tables are processed.

INCLUDE *table-name*
Specifies the table or tables that are to be loaded into the accelerator. If multiple accelerators are specified in the statement, each accelerator is searched for the specified table or tables.

table-name
The qualified name of the table (*schema.name*).

If you specify more than one table name, separate each name by a comma.

If you use wildcard characters, you can specify only one table name, and you must enclose the name or schema with a wildcard character in quotation marks. You can use the following wildcard characters:

_
Indicates a single character

%
Indicates a varying length string

*****
Indicates a varying length string

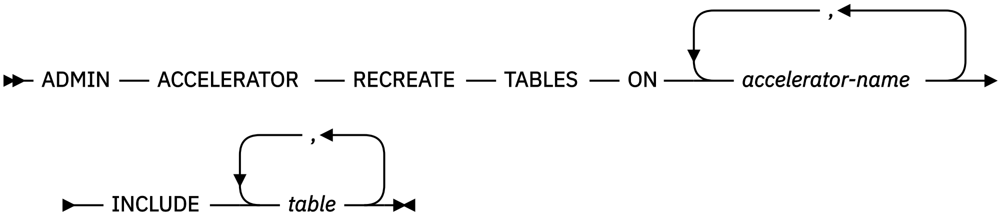
Example table name specifications:

```
ADMF001.TBT1,ADMF002.TBT2
"ADMF%"."TBT_"
SYSADM."*"
```

ADMIN ACCELERATOR RECREATE statement

The ADMIN ACCELERATOR RECREATE statement removes and adds Db2 tables on the accelerators when these tables are dropped and recreated on a Db2 subsystem during an object comparison operation.

Syntax



Option descriptions

accelerator-name
The name of the accelerator on which you want to recreate the specified table or tables.

You can specify more than one accelerator. Separate each name with a comma.

INCLUDE *table*
Specifies the table or tables that are to be recreated on the accelerator. If multiple accelerators are specified in the statement, the table or tables are recreated on each accelerator.

table-name
The qualified name of the table (*schema.name*).

If you specify more than one table name, separate each name by a comma.

If you use wildcard characters, you can specify only one table name, and you must enclose the name or schema with a wildcard character in quotation marks. You can use the following wildcard characters:

_
Indicates a single character

%
Indicates a varying length string

*****
Indicates a varying length string

Example table name specifications:

```
ADMF001.TBT1,ADMF002.TBT2
"ADMF%"."TBT_"
SYSADM."*"
```

Managing Db2 subsystem parameters

Db2 *subsystem parameters* are settings that apply to the entire Db2 for z/OS subsystem. You can view, update, and load these parameters from Db2 Admin Tool.

Tip: To quickly display subsystem parameters from any panel in Db2 Admin Tool, issue the following command:

```
TSO ADBEDIAG ZPARM
```

Before you begin

Managing Db2 subsystem parameters from Db2 Admin Tool requires that Db2 SDSNLOAD data sets be allocated in LINKLIST or STEPLIB. If you do not allocate Db2 SDSNLOAD data sets, you must use the DSNTIJUZ batch job process to assemble and linkedit the DSNZPARM module.

Restriction: Only dynamic parameters can be loaded using this feature.

About this task

Db2 Admin Tool displays the currently active Db2 subsystem parameters and allows you to customize them for your environment. The changed parameters are stored as a new source, which you can then assemble and link-edit into a new DSNZPARM load module. The system parameter source and load modules are referred to by the name DSNZPARM, although you can assign them your own names.

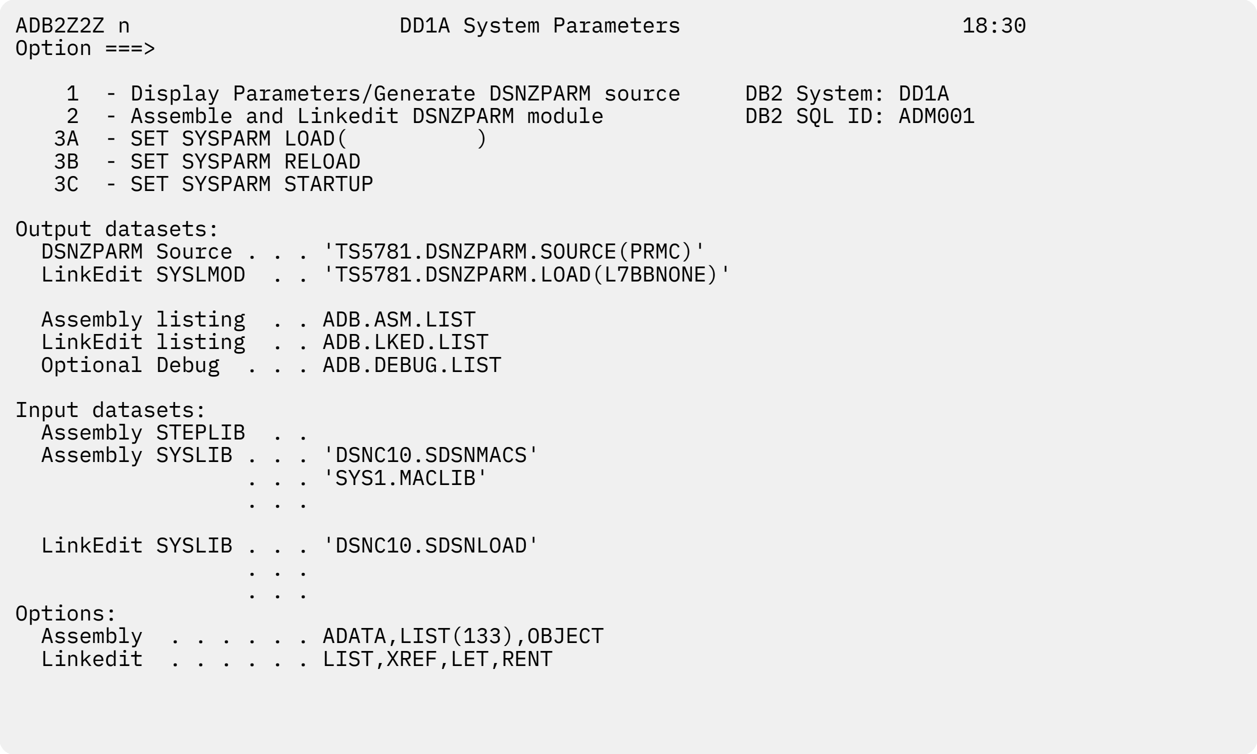
With Db2, you can load a new subsystem parameter module into storage while Db2 is active, which enables you to change certain operational parameters without stopping and starting Db2.

Procedure

To manage Db2 subsystem parameters:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option 2Z, and press Enter.

   The **System Parameters (ADB2Z2Z)** panel is displayed:



```
ADB2Z2Z n                     DD1A System Parameters                      18:30
Option ===>

    1  - Display Parameters/Generate DSNZPARM source     DB2 System: DD1A
    2  - Assemble and Linkedit DSNZPARM module           DB2 SQL ID: ADM001
   3A  - SET SYSPARM LOAD(          )
   3B  - SET SYSPARM RELOAD
   3C  - SET SYSPARM STARTUP

Output datasets:
  DSNZPARM Source . . . 'TS5781.DSNZPARM.SOURCE(PRMC)'
  LinkEdit SYSLMOD  . . 'TS5781.DSNZPARM.LOAD(L7BBNONE)'

  Assembly listing  . . ADB.ASM.LIST
  LinkEdit listing  . . ADB.LKED.LIST
  Optional Debug  . . . ADB.DEBUG.LIST

Input datasets:
  Assembly STEPLIB  . .
  Assembly SYSLIB . . . 'DSNC10.SDSNMACS'
                  . . . 'SYS1.MACLIB'
                  . . .

  LinkEdit SYSLIB . . . 'DSNC10.SDSNLOAD'
                  . . .
                  . . .
Options:
  Assembly  . . . . . . ADATA,LIST(133),OBJECT
  Linkedit  . . . . . . LIST,XREF,LET,RENT
```

*Figure 497. **System Parameters (ADB2Z2Z)** panel*

3. Take one of the following actions:

   - **To display the current parameters and optionally make changes:** Specify option 1 (**Display Parameters/Generate DSNZPARM source**) and specify an output data set and member in the **DSNZPARM Source** field. This data set is where the new source is to be written if you make any changes.
   - **To assemble and link-edit a new parameter source file:** Specify option 2 (**Assemble and Linkedit DSNZPARM module**) and the following information:

- In the **LinkEdit SYSLMOD** field, specify the output SYSLMOD data set name. This data set is where the new load module is stored.
- In the **Input data sets** field, enter information pertaining to additional input libraries and data sets to be used in the assembly and link-edit steps. In the **Assembly SYSLIB** field, specify the data set that contains the DSNZPARM macros, such as DSN6SPRM and DSN6LOGP. Db2 Admin Tool accesses these macros to determine which parameters exist for the subsystem.
- In the **Options** field, specify any options that you want in effect at assembly and link-edit time.

- **To load and activate a new subsystem parameter module:** Specify one of the following SET SYSPARM options:

  3A (SET SYSPARM LOAD)
  Loads the specified load module, which is DSNZPARM by default.

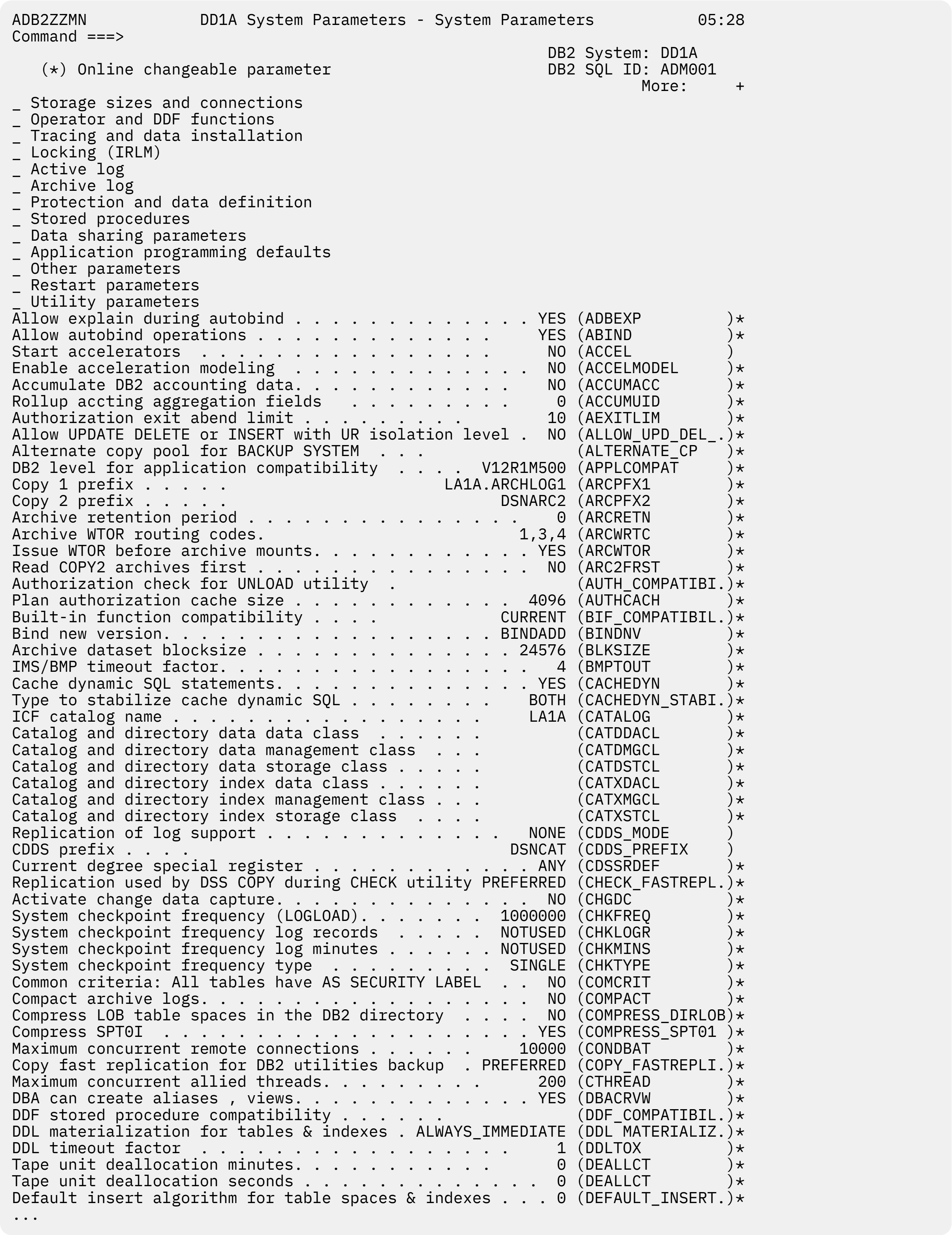
  3B (SET SYSPARM RELOAD)
  Reloads the previous subsystem parameter load module.

  3C (SET SYSPARM STARTUP)
  Reloads the parameter load module used at subsystem startup.

The System Parameters — System Parameters (ADB2ZZMN) panel

The **System Parameters — System Parameters (ADB2ZZMN)** panel displays a list of current values for Db2 subsystem parameters (zparms). From this panel, you can also change the parameter values.

To open the **System Parameters — System Parameters (ADB2ZZMN)** panel, select option 1 on the **System Parameters (ADB2Z2Z)** panel.



```
ADB2ZZMN            DD1A System Parameters - System Parameters           05:28
Command ===>
                                                         DB2 System: DD1A
   (*) Online changeable parameter                       DB2 SQL ID: ADM001
                                                                   More:     +
_ Storage sizes and connections
_ Operator and DDF functions
_ Tracing and data installation
_ Locking (IRLM)
_ Active log
_ Archive log
_ Protection and data definition
_ Stored procedures
_ Data sharing parameters
_ Application programming defaults
_ Other parameters
_ Restart parameters
_ Utility parameters
Allow explain during autobind . . . . . . . . . . . . . . YES (ADBEXP         )*
Allow autobind operations . . . . . . . . . . . . .     YES (ABIND          )*
Start accelerators  . . . . . . . . . . . . . . . .      NO (ACCEL          )
Enable acceleration modeling  . . . . . . . . . . . . .  NO (ACCELMODEL     )*
Accumulate DB2 accounting data. . . . . . . . . . .    NO (ACCUMACC       )*
Rollup accting aggregation fields   . . . . . . . . .     0 (ACCUMUID       )*
Authorization exit abend limit . . . . . . . .         10 (AEXITLIM       )*
Allow UPDATE DELETE or INSERT with UR isolation level .  NO (ALLOW_UPD_DEL_.)*
Alternate copy pool for BACKUP SYSTEM  . . .                (ALTERNATE_CP   )*
DB2 level for application compatibility  . . . .  V12R1M500 (APPLCOMPAT     )*
Copy 1 prefix . . . . .                       LA1A.ARCHLOG1 (ARCPFX1        )*
Copy 2 prefix . . . . .                             DSNARC2 (ARCPFX2        )*
Archive retention period . . . . . . . . . . . . . . .    0 (ARCRETN        )*
Archive WTOR routing codes.                           1,3,4 (ARCWRTC        )*
Issue WTOR before archive mounts. . . . . . . . . . . . YES (ARCWTOR        )*
Read COPY2 archives first . . . . . . . . . . . . . . .  NO (ARC2FRST       )*
Authorization check for UNLOAD utility  .                   (AUTH_COMPATIBI.)*
Plan authorization cache size . . . . . . . . . . . .  4096 (AUTHCACH       )*
Built-in function compatibility . . . .             CURRENT (BIF_COMPATIBIL.)*
Bind new version. . . . . . . . . . . . . . . . . . BINDADD (BINDNV         )*
Archive dataset blocksize . . . . . . . . . . . . . . 24576 (BLKSIZE        )*
IMS/BMP timeout factor. . . . . . . . . . . . . . . .    4 (BMPTOUT        )*
Cache dynamic SQL statements. . . . . . . . . . . . . YES (CACHEDYN       )*
Type to stabilize cache dynamic SQL . . . . . . . .    BOTH (CACHEDYN_STABI.)*
ICF catalog name . . . . . . . . . . . . . . . . .     LA1A (CATALOG        )*
Catalog and directory data data class  . . . . . .          (CATDDACL       )*
Catalog and directory data management class  . . .          (CATDMGCL       )*
Catalog and directory data storage class . . . . .          (CATDSTCL       )*
Catalog and directory index data class . . . . . .          (CATXDACL       )*
Catalog and directory index management class . . .          (CATXMGCL       )*
Catalog and directory index storage class  . . . .          (CATXSTCL       )*
Replication of log support . . . . . . . . . . . . .   NONE (CDDS_MODE      )
CDDS prefix . . . .                                  DSNCAT (CDDS_PREFIX    )
Current degree special register . . . . . . . . . . . . ANY (CDSSRDEF       )*
Replication used by DSS COPY during CHECK utility PREFERRED (CHECK_FASTREPL.)*
Activate change data capture. . . . . . . . . . . . . .  NO (CHGDC          )*
System checkpoint frequency (LOGLOAD). . . . . .  1000000 (CHKFREQ        )*
System checkpoint frequency log records  . . . . .  NOTUSED (CHKLOGR        )*
System checkpoint frequency log minutes . . . . . . NOTUSED (CHKMINS        )*
System checkpoint frequency type  . . . . . . . . .  SINGLE (CHKTYPE        )*
Common criteria: All tables have AS SECURITY LABEL  . .  NO (COMCRIT        )*
Compact archive logs. . . . . . . . . . . . . . . . .  NO (COMPACT        )*
Compress LOB table spaces in the DB2 directory  . . . .  NO (COMPRESS_DIRLOB)*
Compress SPT0I  . . . . . . . . . . . . . . . . . . . . YES (COMPRESS_SPT01 )*
Maximum concurrent remote connections . . . . . .     10000 (CONDBAT        )*
Copy fast replication for DB2 utilities backup  . PREFERRED (COPY_FASTREPLI.)*
Maximum concurrent allied threads. . . . . . . .      200 (CTHREAD        )*
DBA can create aliases , views. . . . . . . . . . . .  YES (DBACRVW        )*
DDF stored procedure compatibility . . . . . .              (DDF_COMPATIBIL.)*
DDL materialization for tables & indexes . ALWAYS_IMMEDIATE (DDL MATERIALIZ.)*
DDL timeout factor  . . . . . . . . . . . . . . . . .     1 (DDLTOX         )*
Tape unit deallocation minutes. . . . . . . . . .      0 (DEALLCT        )*
Tape unit deallocation seconds . . . . . . . . . . . . .  0 (DEALLCT        )*
Default insert algorithm for table spaces & indexes . . . 0 (DEFAULT_INSERT.)*
...
```

Figure 498. ***System Parameters — System Parameters (ADB2ZZMN)*** *panel*

On this panel, the first 13 lines, which have no parameter values, are selection fields. When selected, a secondary panel is displayed that shows the parameters organized by category.

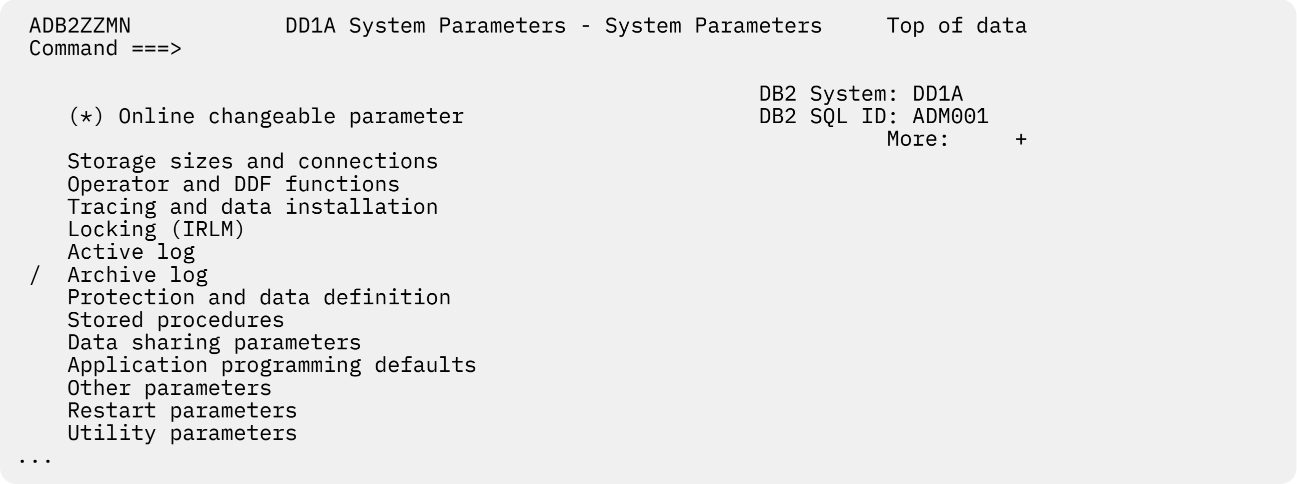
The selection fields are followed by the dynamic parameters in alphabetical order. Enter new values for any parameters by overwriting the existing value. Only those parameters identified by an asterisk (*) can be loaded dynamically using the SET SYSPARM command.

Restriction: A message might be issued for parameters not on this panel, but whose value has changed as a result of the assembly. This situation might occur if Db2 maintenance was applied to the macro data sets, thereby changing the internal parameter values, and no interim subsystem recycle was performed.

System Parameters - Archive Log (ADB2ZZAL) panel

The **System Parameters - Archive Log (ADB2ZZAL)** panel is an example of a secondary system parameter panel that is displayed when one of the fields is selected on the **System Parameters — System Parameters (ADB2ZZMN)** panel.

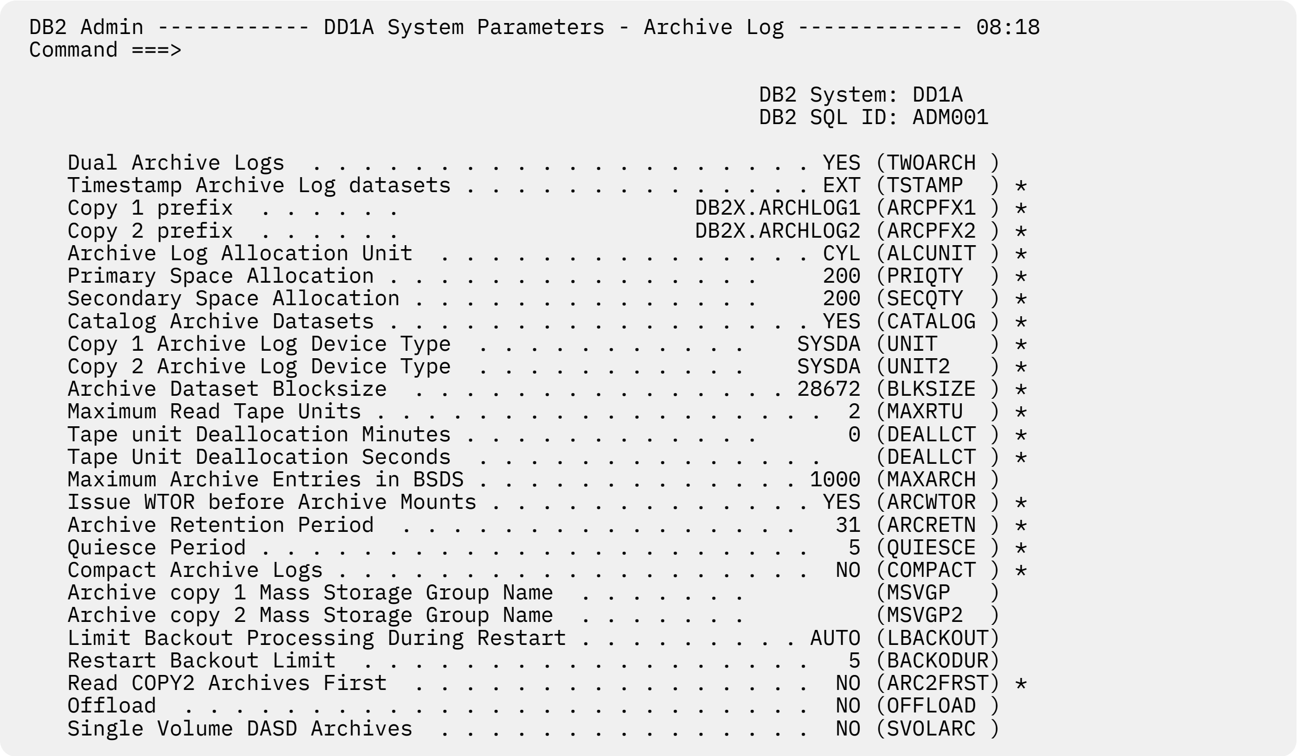
In this case, the **Archive Log** field is selected on the **System Parameters — System Parameters (ADB2ZZMN)** panel:



```
 ADB2ZZMN            DD1A System Parameters - System Parameters     Top of data
 Command ===>

                                                         DB2 System: DD1A
    (*) Online changeable parameter                      DB2 SQL ID: ADM001
                                                                   More:     +
    Storage sizes and connections
    Operator and DDF functions
    Tracing and data installation
    Locking (IRLM)
    Active log
 /  Archive log
    Protection and data definition
    Stored procedures
    Data sharing parameters
    Application programming defaults
    Other parameters
    Restart parameters
    Utility parameters
...
```

*Figure 499. **System Parameters — System Parameters (ADB2ZZMN)** panel with **Archive Log** selected*

As a result, the **System Parameters - Archive Log (ADB2ZZAL)** panel is displayed:



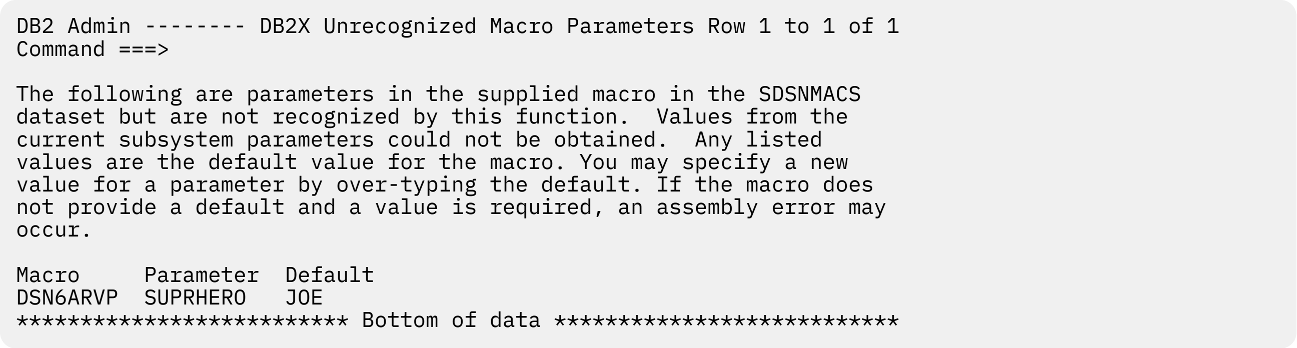
```
DB2 Admin ------------ DD1A System Parameters - Archive Log ------------- 08:18
Command ===>

                                                           DB2 System: DD1A
                                                           DB2 SQL ID: ADM001

   Dual Archive Logs  . . . . . . . . . . . . . . . . . . . . YES (TWOARCH )
   Timestamp Archive Log datasets . . . . . . . . . . . . . . EXT (TSTAMP  ) *
   Copy 1 prefix  . . . . . .                       DB2X.ARCHLOG1 (ARCPFX1 ) *
   Copy 2 prefix  . . . . . .                       DB2X.ARCHLOG2 (ARCPFX2 ) *
   Archive Log Allocation Unit  . . . . . . . . . . . . . . . CYL (ALCUNIT ) *
   Primary Space Allocation . . . . . . . . . . . . . . .     200 (PRIQTY  ) *
   Secondary Space Allocation . . . . . . . . . . . . . .     200 (SECQTY  ) *
   Catalog Archive Datasets . . . . . . . . . . . . . . . . . YES (CATALOG ) *
   Copy 1 Archive Log Device Type  . . . . . . . . . . .    SYSDA (UNIT    ) *
   Copy 2 Archive Log Device Type  . . . . . . . . . . .    SYSDA (UNIT2   ) *
   Archive Dataset Blocksize  . . . . . . . . . . . . . . . 28672 (BLKSIZE ) *
   Maximum Read Tape Units . . . . . . . . . . . . . . . . . .  2 (MAXRTU  ) *
   Tape unit Deallocation Minutes . . . . . . . . . . . .       0 (DEALLCT ) *
   Tape Unit Deallocation Seconds  . . . . . . . . . . . . . .    (DEALLCT ) *
   Maximum Archive Entries in BSDS . . . . . . . . . . . . . 1000 (MAXARCH )
   Issue WTOR before Archive Mounts . . . . . . . . . . . . . YES (ARCWTOR ) *
   Archive Retention Period  . . . . . . . . . . . . . . . .   31 (ARCRETN ) *
   Quiesce Period . . . . . . . . . . . . . . . . . . . . . .   5 (QUIESCE ) *
   Compact Archive Logs . . . . . . . . . . . . . . . . . . .  NO (COMPACT ) *
   Archive copy 1 Mass Storage Group Name  . . . . . . .          (MSVGP   )
   Archive copy 2 Mass Storage Group Name  . . . . . . .          (MSVGP2  )
   Limit Backout Processing During Restart . . . . . . . . . AUTO (LBACKOUT)
   Restart Backout Limit  . . . . . . . . . . . . . . . . . .   5 (BACKODUR)
   Read COPY2 Archives First  . . . . . . . . . . . . . . . .  NO (ARC2FRST) *
   Offload  . . . . . . . . . . . . . . . . . . . . . . . . .  NO (OFFLOAD )
   Single Volume DASD Archives  . . . . . . . . . . . . . . .  NO (SVOLARC )
```

*Figure 500. **System Parameters - Archive Log (ADB2ZZAL)** panel*

Unrecognized Macro Parameters panel

Db2 Admin Tool accesses SDSNMACS, the Assembly SYSLIB data set specified by the user, to determine which DSNZPARM parameters exist for this subsystem.

An unrecognized macro was encountered and is displayed in the **Unrecognized Macro Parameters** panel, as shown in the following figure.



```
DB2 Admin -------- DB2X Unrecognized Macro Parameters Row 1 to 1 of 1
Command ===>

The following are parameters in the supplied macro in the SDSNMACS
dataset but are not recognized by this function.  Values from the
current subsystem parameters could not be obtained.  Any listed
values are the default value for the macro. You may specify a new
value for a parameter by over-typing the default. If the macro does
not provide a default and a value is required, an assembly error may
occur.

Macro     Parameter  Default
DSN6ARVP  SUPRHERO   JOE
************************** Bottom of data ***************************
```

Figure 501. Unrecognized Macro Parameters panel (ADB2ZZTL)

Fast index traversal

Fast index traversal is a feature of Db2 that can improve overall random index access. You can manage this feature from the **Manage Fast Index Traversal (ADB2Z2I)** panel.

Related information

Fast index traversal (Db2 12 for z/OS)

Viewing memory used for fast index traversal

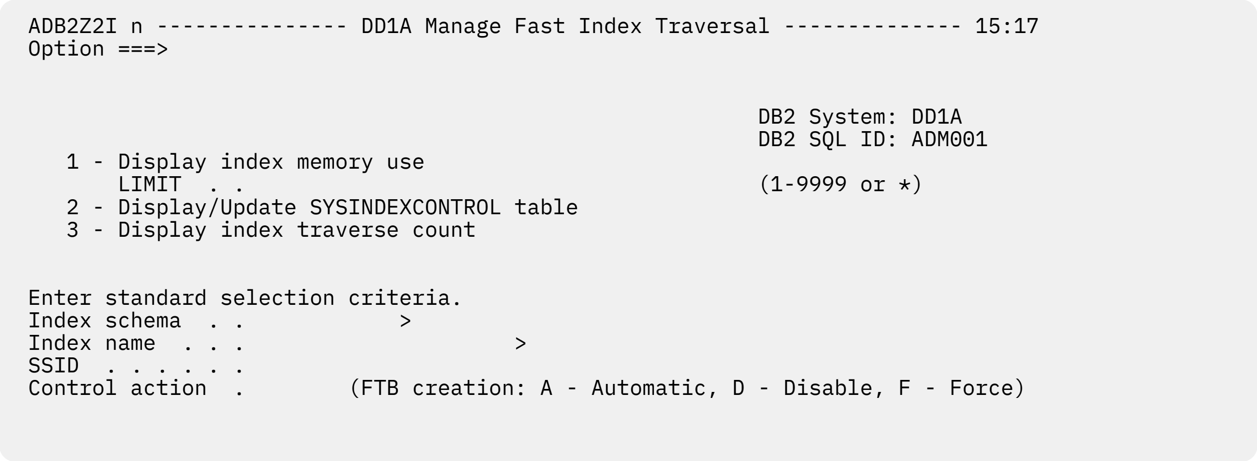
You can view statistics about the current amount of memory that Db2 uses for fast index traversal.

Procedure

To view memory used for fast index traversal:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option 2I, and press Enter.
3. On the **Manage Fast Index Traversal (ADB2Z2I)** panel, specify option 1.

   Tip: To limit the number of output messages displayed, also specify a value in the **LIMIT** field.



```
ADB2Z2I n --------------- DD1A Manage Fast Index Traversal -------------- 15:17
Option ===>



                                                         DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
   1 - Display index memory use
       LIMIT  . .                                        (1-9999 or *)
   2 - Display/Update SYSINDEXCONTROL table
   3 - Display index traverse count


Enter standard selection criteria.
Index schema  . .            >
Index name  . . .                     >
SSID  . . . . . .
Control action  .        (FTB creation: A - Automatic, D - Disable, F - Force)
```

Figure 502. ***Manage Fast Index Traversal (ADB2Z2I)*** *panel*

4. Press Enter.

   The Db2 command `-DIS STATS(INDEXMEMORYUSAGE)` is issued, and the requested memory statistics are displayed.

Related information

-DISPLAY STATS (Db2) (Db2 12 for z/OS documentation)

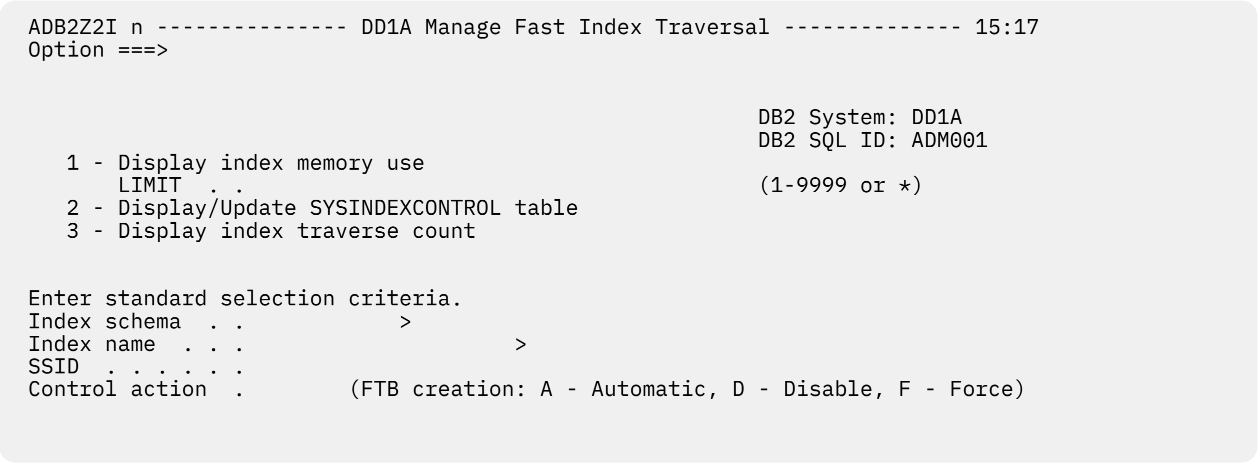
Setting the rules for fast index traversal

The rules that determine which indexes or index partitions use fast index traversal are set in the Db2 catalog table SYSINDEXCONTROL. To set or change these rules, you must modify rows in this table.

Procedure

To set the rules for fast index traversal:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option 2I, and press Enter.
3. On the **Manage Fast Index Traversal (ADB2Z2I)** panel, specify option 2, and press Enter.

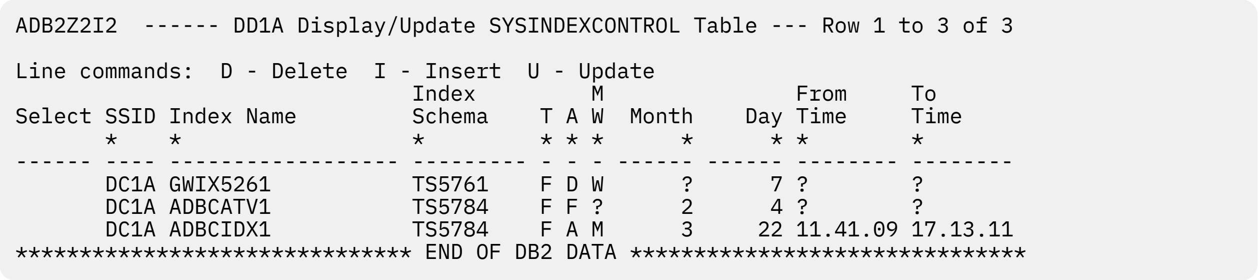


```
ADB2Z2I n --------------- DD1A Manage Fast Index Traversal ------------- 15:17
Option ===>


                                                         DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
   1 - Display index memory use
       LIMIT  . .                                        (1-9999 or *)
   2 - Display/Update SYSINDEXCONTROL table
   3 - Display index traverse count


Enter standard selection criteria.
Index schema  . .            >
Index name  . . .                     >
SSID  . . . . . .
Control action  .        (FTB creation: A - Automatic, D - Disable, F - Force)
```

*Figure 503. **Manage Fast Index Traversal (ADB2Z2I)** panel*

4. On the **Display/Update SYSINDEXCONTROL Table (ADB2Z2I2)** panel, use the D I, and U line commands to modify the SYSINDEXCONTROL table.

   The entries in this table control which indexes or index partitions use fast index traversal. For detailed information on how to modify SYSINDEXCONTROL, see Enabling or disabling fast index traversal at the index level (Db2 12 for z/OS).



```
ADB2Z2I2  ------ DD1A Display/Update SYSINDEXCONTROL Table --- Row 1 to 3 of 3

Line commands:  D - Delete  I - Insert  U - Update
                               Index         M               From     To
Select SSID Index Name         Schema    T A W  Month    Day Time     Time
       *    *                  *         * * *      *      * *        *
------ ---- ------------------ --------- - - - ------ ------ -------- --------
       DC1A GWIX5261           TS5761    F D W      ?      7 ?        ?
       DC1A ADBCATV1           TS5784    F F ?      2      4 ?        ?
       DC1A ADBCIDX1           TS5784    F A M      3     22 11.41.09 17.13.11
******************************* END OF DB2 DATA *******************************
```

Viewing index traverse counts

You can view the index traverse counts for a specific index or for a specified number of indexes with the highest traverse counts in descending order.

Tip: As an alternative to this procedure, you can use the DISITC line command from any of the following panels:

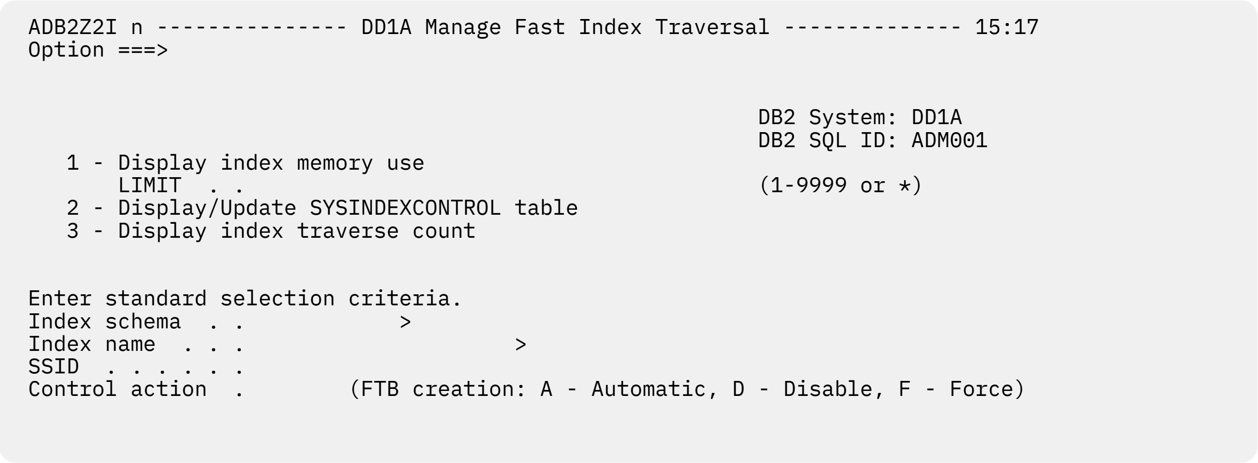
- **Databases (ADB21D)** panel
- **Indexes (ADB21X)** panel
- **Index Parts (ADB21XP)** panel

Procedure

To view index traverse counts:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option 2I, and press Enter.
3. On the **Manage Fast Index Traversal (ADB2Z2I)** panel, specify option 3.

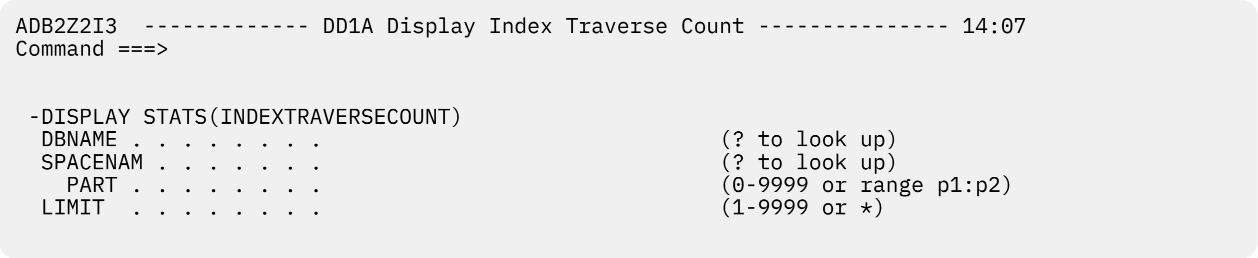
   Tip: To limit the number of output messages displayed, also specify a value in the **LIMIT** field.



```
ADB2Z2I n --------------- DD1A Manage Fast Index Traversal -------------- 15:17
Option ===>


                                                         DB2 System: DD1A
                                                         DB2 SQL ID: ADM001
   1 - Display index memory use
       LIMIT  . .                                        (1-9999 or *)
   2 - Display/Update SYSINDEXCONTROL table
   3 - Display index traverse count


Enter standard selection criteria.
Index schema  . .            >
Index name   . . .                    >
SSID  . . . . . .
Control action   .        (FTB creation: A - Automatic, D - Disable, F - Force)
```

*Figure 504. **Manage Fast Index Traversal (ADB2Z2I)** panel*

4. Press Enter.
5. On the **Display Index Traverse Count (ADB2Z2I3)** panel, specify any additional parameters:



```
ADB2Z2I3  ------------- DD1A Display Index Traverse Count --------------- 14:07
Command ===>


 -DISPLAY STATS(INDEXTRAVERSECOUNT)
  DBNAME . . . . . . . .                                 (? to look up)
  SPACENAM . . . . . . .                                 (? to look up)
    PART . . . . . . . .                                 (0-9999 or range p1:p2)
  LIMIT  . . . . . . . .                                 (1-9999 or *)
```

6. Press Enter.

   The Db2 command DIS STATS(INDEXTRAVERSECOUNT) is issued, and the requested counts are displayed.

Related information

-DISPLAY STATS (Db2) (Db2 12 for z/OS documentation)

Displaying global variables and their authorizations

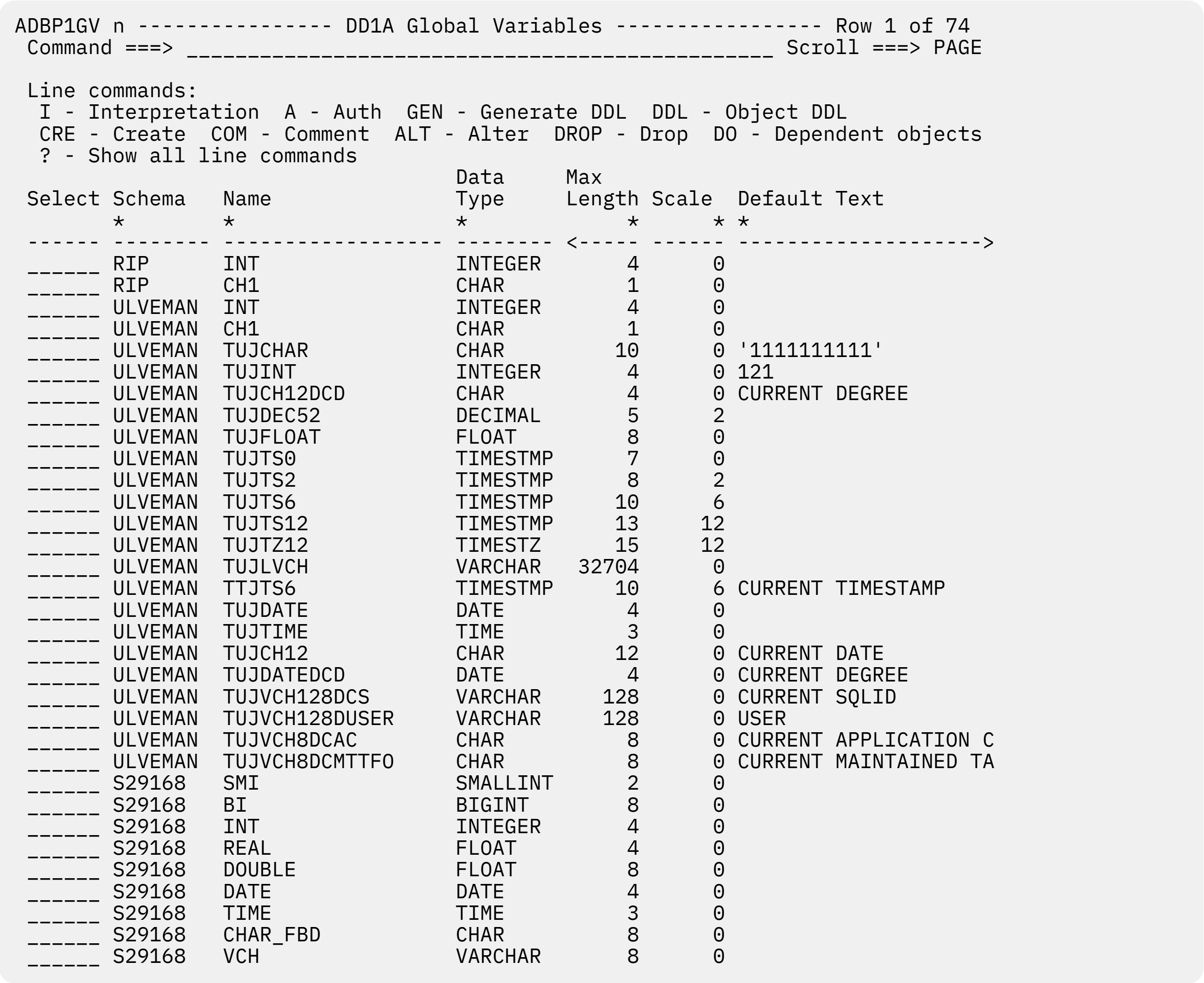
Global variables enable you to share relational data between SQL statements without the need for application logic to support the data transfer. Information about global variables is stored in the SYSVARIABLES catalog table.

Procedure

To display global variables:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify option GV, and press Enter.

   The **Global Variables (ADBP1GV)** panel is displayed, as shown in the following figure:



```
ADBP1GV n ---------------- DD1A Global Variables ----------------- Row 1 of 74
 Command ===> ________________________________________________ Scroll ===> PAGE

 Line commands:
  I - Interpretation  A - Auth  GEN - Generate DDL  DDL - Object DDL
  CRE - Create  COM - Comment  ALT - Alter  DROP - Drop  DO - Dependent objects
  ? - Show all line commands
                                    Data     Max
 Select Schema   Name               Type     Length Scale  Default Text
        *        *                  *             *      * *
 ------ -------- ------------------ -------- <----- ------ -------------------->
 ______ RIP      INT                INTEGER       4      0
 ______ RIP      CH1                CHAR          1      0
 ______ ULVEMAN  INT                INTEGER       4      0
 ______ ULVEMAN  CH1                CHAR          1      0
 ______ ULVEMAN  TUJCHAR            CHAR         10      0 '1111111111'
 ______ ULVEMAN  TUJINT             INTEGER       4      0 121
 ______ ULVEMAN  TUJCH12DCD         CHAR          4      0 CURRENT DEGREE
 ______ ULVEMAN  TUJDEC52           DECIMAL       5      2
 ______ ULVEMAN  TUJFLOAT           FLOAT         8      0
 ______ ULVEMAN  TUJTS0             TIMESTMP      7      0
 ______ ULVEMAN  TUJTS2             TIMESTMP      8      2
 ______ ULVEMAN  TUJTS6             TIMESTMP     10      6
 ______ ULVEMAN  TUJTS12            TIMESTMP     13     12
 ______ ULVEMAN  TUJTZ12            TIMESTZ      15     12
 ______ ULVEMAN  TUJLVCH            VARCHAR   32704      0
 ______ ULVEMAN  TTJTS6             TIMESTMP     10      6 CURRENT TIMESTAMP
 ______ ULVEMAN  TUJDATE            DATE          4      0
 ______ ULVEMAN  TUJTIME            TIME          3      0
 ______ ULVEMAN  TUJCH12            CHAR         12      0 CURRENT DATE
 ______ ULVEMAN  TUJDATEDCD         DATE          4      0 CURRENT DEGREE
 ______ ULVEMAN  TUJVCH128DCS       VARCHAR     128      0 CURRENT SQLID
 ______ ULVEMAN  TUJVCH128DUSER     VARCHAR     128      0 USER
 ______ ULVEMAN  TUJVCH8DCAC        CHAR          8      0 CURRENT APPLICATION C
 ______ ULVEMAN  TUJVCH8DCMTTFO     CHAR          8      0 CURRENT MAINTAINED TA
 ______ S29168   SMI                SMALLINT      2      0
 ______ S29168   BI                 BIGINT        8      0
 ______ S29168   INT                INTEGER       4      0
 ______ S29168   REAL               FLOAT         4      0
 ______ S29168   DOUBLE             FLOAT         8      0
 ______ S29168   DATE               DATE          4      0
 ______ S29168   TIME               TIME          3      0
 ______ S29168   CHAR_FBD           CHAR          8      0
 ______ S29168   VCH                VARCHAR       8      0
```

Figure 505. ***Global Variables (ADBP1GV)*** *panel*

The following fields are displayed on this panel:

Schema
The schema of the global variable.

Name
The name of the global variable.

Data Type
The name of the data type.

Max Length
The maximum length of the global variable.

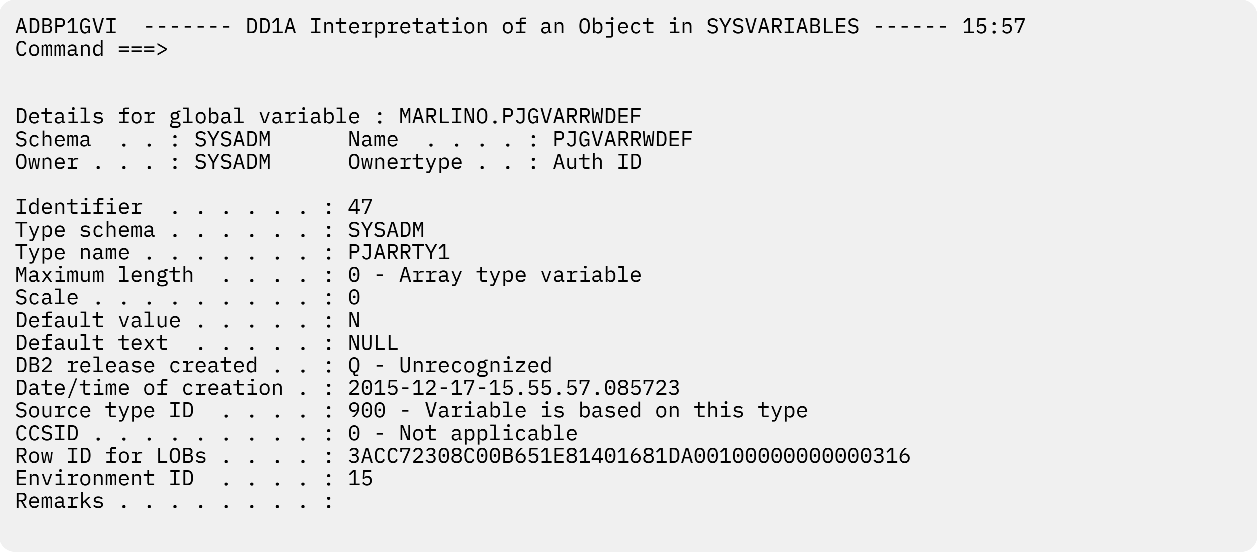
Scale
The scale of the global variable.

Default Text
The text of the default value of the global variable.

If the text is truncated, type EXPAND on the primary command line, position the cursor on the default text field, and press Enter to display all of the text.

3. Issue the I line command next to the global variable about which you want more information, and press Enter.

The **Interpretation of an Object in SYSVARIABLES (ADBP1GVI)** panel is displayed, as shown in the following figure:



```
ADBP1GVI  ------- DD1A Interpretation of an Object in SYSVARIABLES ------ 15:57
Command ===>


Details for global variable : MARLINO.PJGVARRWDEF
Schema  . . : SYSADM      Name  . . . . : PJGVARRWDEF
Owner . . . : SYSADM      Ownertype . . : Auth ID

Identifier  . . . . . . : 47
Type schema . . . . . . : SYSADM
Type name . . . . . . . : PJARRTY1
Maximum length  . . . . : 0 - Array type variable
Scale . . . . . . . . . : 0
Default value . . . . . : N
Default text  . . . . . : NULL
DB2 release created . . : Q - Unrecognized
Date/time of creation . : 2015-12-17-15.55.57.085723
Source type ID  . . . . : 900 - Variable is based on this type
CCSID . . . . . . . . . : 0 - Not applicable
Row ID for LOBs . . . . : 3ACC72308C00B651E81401681DA00100000000000316
Environment ID  . . . . : 15
Remarks . . . . . . . . :
```

Figure 506. ***Interpretation of an Object in SYSVARIABLES (ADBP1GVI)*** *panel*

The following fields are displayed on this panel:

Schema
The schema of the global variable.

Name
The name of the global variable.

Owner
The authorization ID of the owner of the global variable.

Ownertype
The type of owner:

L
The owner is a role.

blank
The owner is an authorization ID.

Type schema
The schema name of the data type. For built-in data types, this value is SYSIBM.

Type name
The unqualified name of the data type.

Maximum length
The maximum length of the global variable.

Scale
The scale of the global variable.

Default text
The text of the default value of the global variable.

If the text is truncated, type EXPAND on the primary command line, position the cursor on the default text field, and press Enter to display all of the text.

Identifier
The identifier of the global variable.

DB2 release created
The release of Db2 that was used to create the object.

Date/time of creation
The date and time that the global variable was created.

Source type
The source type:

0
A built-in data type.

internal_ID
A distinct type.

CCSID
The CCSID of the global variable. The CCSID encoding scheme and character set.

Default clause
The default clause that is specified for the global variable.

Row ID for LOBs
The row ID values for the LOB columns in the SYSVARIABLES table.

Internal environment
The internal environment identifier.

IBM required
The origin of the row:

Y
The row came from the basic machine-readable material (MRM) tape.

N
The row did not come from the basic machine-readable material (MRM) tape.

Displaying global variable authorizations

You can display information about the users who grant privileges to global variables, and information about the users who hold the privileges. You can also display information about any plans that use the privileges.

About this task

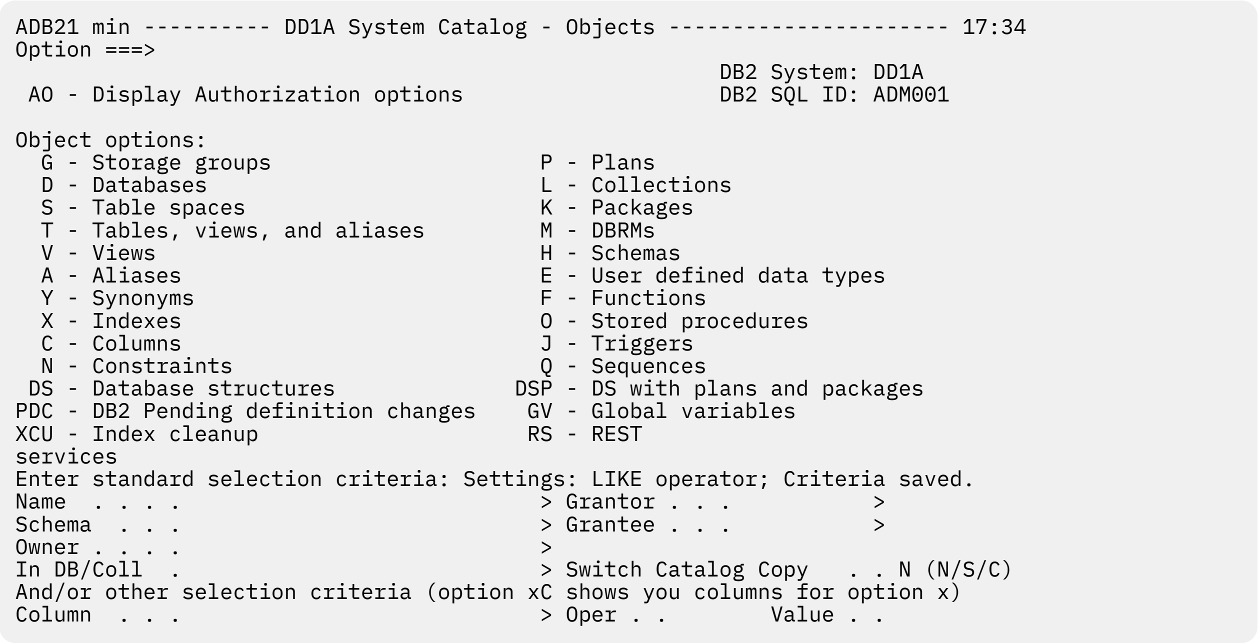
Authorization information is stored in the SYSIBM.SYSVARIABLEAUTH catalog table.

Procedure

To display global variable authorizations:

1. On the **System Catalog (ADB21)** panel, specify option A0, and press Enter.

   Authorization options are displayed on the **System Catalog (ADB21)** panel - Authorization options, as shown in the following figure.

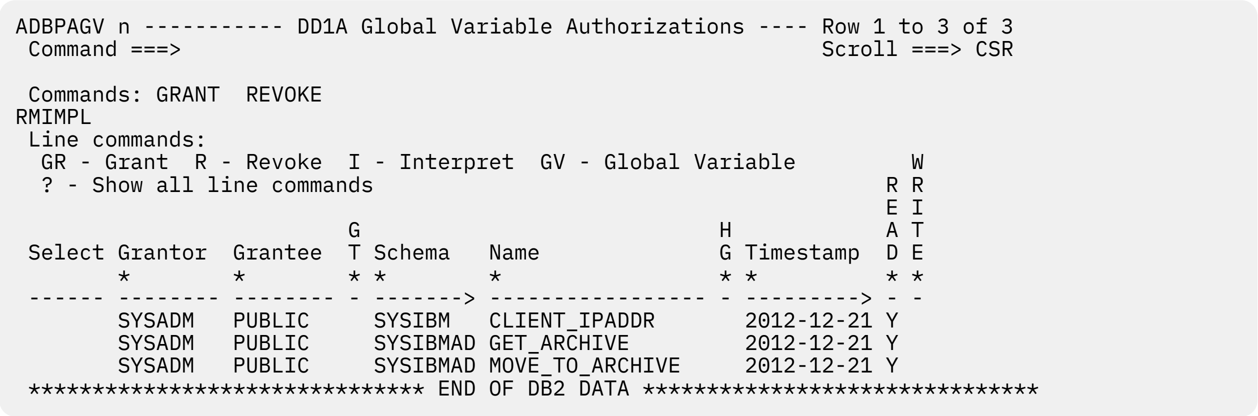


```
ADB21 min ---------- DD1A System Catalog - Objects --------------------- 17:34
Option ===>
                                                       DB2 System: DD1A
 AO - Display Authorization options                    DB2 SQL ID: ADM001

Object options:
  G - Storage groups                     P - Plans
  D - Databases                          L - Collections
  S - Table spaces                       K - Packages
  T - Tables, views, and aliases         M - DBRMs
  V - Views                              H - Schemas
  A - Aliases                            E - User defined data types
  Y - Synonyms                           F - Functions
  X - Indexes                            O - Stored procedures
  C - Columns                            J - Triggers
  N - Constraints                        Q - Sequences
 DS - Database structures              DSP - DS with plans and packages
PDC - DB2 Pending definition changes    GV - Global variables
XCU - Index cleanup                     RS - REST
services
Enter standard selection criteria: Settings: LIKE operator; Criteria saved.
Name  . . . .                            > Grantor . . .           >
Schema  . . .                            > Grantee . . .           >
Owner . . . .                            >
In DB/Coll  .                            > Switch Catalog Copy   . . N (N/S/C)
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .                            > Oper . .        Value . .
```

*Figure 507. **System Catalog (ADB21)** panel - using search criteria*

2. Specify option GVA, and press Enter.

   The **Global Variable Authorizations (ADBPAGV)** panel is displayed, as shown in the following figure:



```
ADBPAGV n ----------- DD1A Global Variable Authorizations ---- Row 1 to 3 of 3
 Command ===>                                                  Scroll ===> CSR

 Commands: GRANT  REVOKE
RMIMPL
 Line commands:
  GR - Grant  R - Revoke  I - Interpret  GV - Global Variable         W
  ? - Show all line commands                                        R R
                                                                    E I
                          G                            H            A T
 Select Grantor  Grantee  T Schema   Name              G Timestamp  D E
        *        *        * *        *                 * *          * *
 ------ -------- -------- - -------> ----------------- - ---------> - -
        SYSADM   PUBLIC     SYSIBM   CLIENT_IPADDR       2012-12-21 Y
        SYSADM   PUBLIC     SYSIBMAD GET_ARCHIVE         2012-12-21 Y
        SYSADM   PUBLIC     SYSIBMAD MOVE_TO_ARCHIVE     2012-12-21 Y
 ******************************* END OF DB2 DATA *******************************
```

*Figure 508. **Global Variable Authorizations (ADBPAGV)** panel*

The following fields are displayed on this panel:

Grantor
The authorization ID of the user who granted the privilege.

Grantee
The authorization ID of the user who holds the privilege or the name of the plan that uses the privilege.

GT
Grantee type, which can be one of the following values:

blank
An authorization ID.

L
A role.

P
An application package. The grantee is a package if COLLID is not blank.

Schema
The schema name of the global variable.

Name
The unqualified name of the global variable.

HG
The authorization level of the user who granted the privileges:

blank
Not applicable.

E
SECADM.

G
ACCESSCTRL.

S
SYSADM.

T
DATAACCESS.

Timestamp
The time when the GRANT statement was run.

READ
The privilege to read the global variable:

blank
Not held.

G
Read from GRANT.

Y
Read without GRANT.

WRITE
The privilege to write the global variable:

blank
Not held.

G
Read from GRANT.

Y
Read without GRANT.

3. Issue the I line command for the authorization about which you want more information, and press Enter.

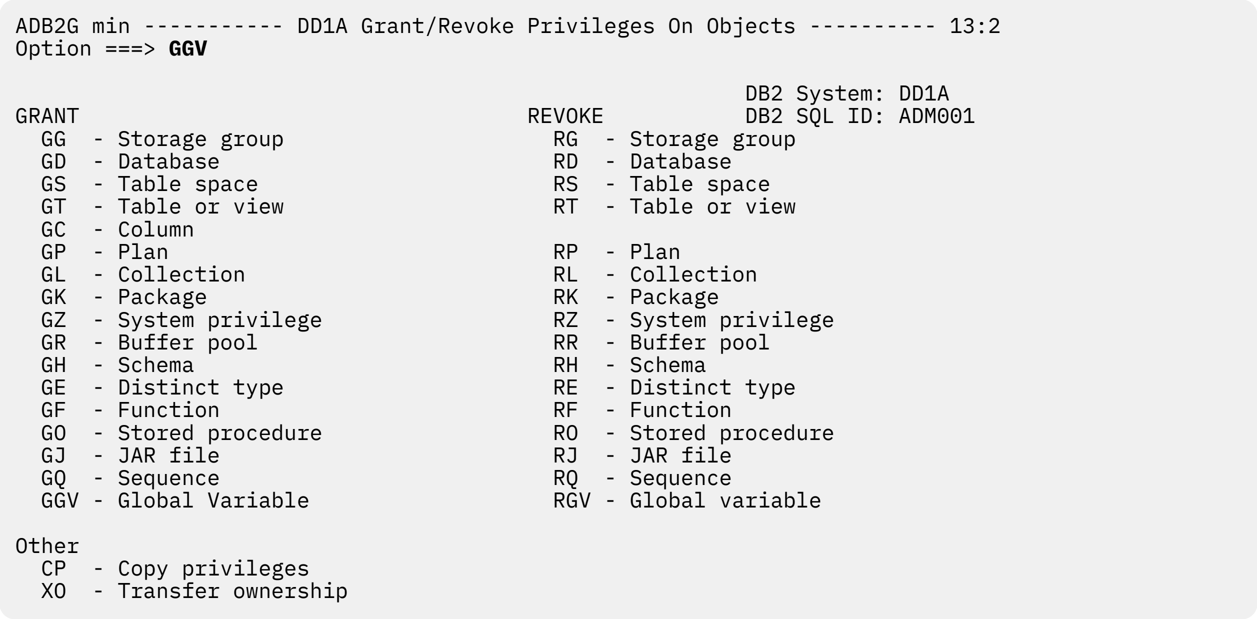
   The **Interpretation of Object in SYSVARIABLEAUTH (ADBPAGVI)** panel is displayed with the requested authorization information.

Granting global variable authorizations

You can grant privileges to users so that they can use global variables. You can also grant the authority to grant privileges to others. Db2 Admin Tool guides you through the process without requiring you to know the syntax of the GRANT SQL statements.

Procedure

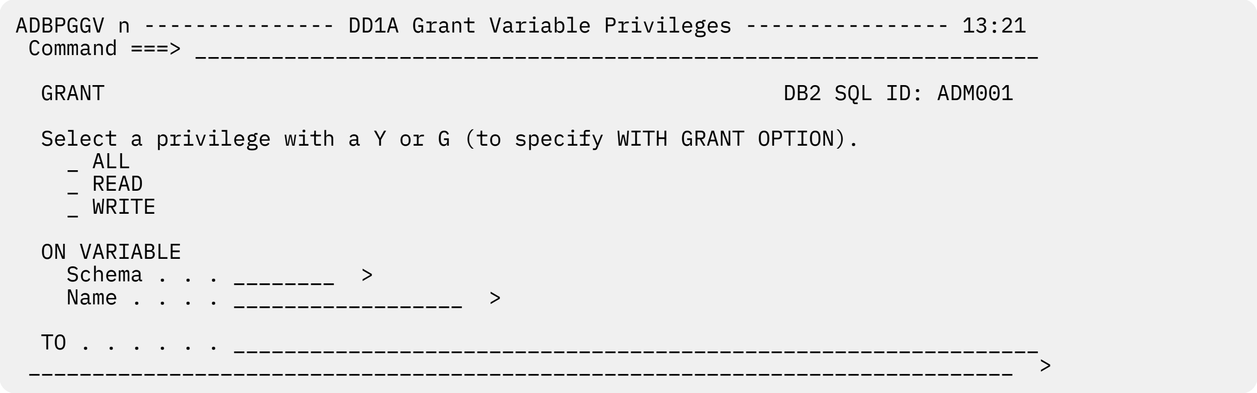
1. On the **DB2 Administration Menu (ADB2)** panel, specify 2 and press Enter.
2. On the **Execute SQL Statements (ADB22)** panel, specify option 5 and press Enter.
3. On the **Grant/Revoke Privileges On Objects (ADB2G)** panel, specify the GGV option and press Enter:



```
ADB2G min ----------- DD1A Grant/Revoke Privileges On Objects ---------- 13:2
Option ===> GGV

                                                         DB2 System: DD1A
GRANT                                   REVOKE           DB2 SQL ID: ADM001
  GG  - Storage group                     RG  - Storage group
  GD  - Database                          RD  - Database
  GS  - Table space                       RS  - Table space
  GT  - Table or view                     RT  - Table or view
  GC  - Column
  GP  - Plan                              RP  - Plan
  GL  - Collection                        RL  - Collection
  GK  - Package                           RK  - Package
  GZ  - System privilege                  RZ  - System privilege
  GR  - Buffer pool                       RR  - Buffer pool
  GH  - Schema                            RH  - Schema
  GE  - Distinct type                     RE  - Distinct type
  GF  - Function                          RF  - Function
  GO  - Stored procedure                  RO  - Stored procedure
  GJ  - JAR file                          RJ  - JAR file
  GQ  - Sequence                          RQ  - Sequence
  GGV - Global Variable                   RGV - Global variable

Other
  CP  - Copy privileges
  XO  - Transfer ownership
```

Figure 509. Grant/Revoke Privileges On Objects panel (ADB2G)

4. On the **Grant Variable Privileges (ADBPGGV)** panel, complete the fields and press Enter to grant the selected privilege.



```
ADBPGGV n --------------- DD1A Grant Variable Privileges ---------------- 13:21
 Command ===> __________________________________________________________________

  GRANT                                                   DB2 SQL ID: ADM001

  Select a privilege with a Y or G (to specify WITH GRANT OPTION).
    _ ALL
    _ READ
    _ WRITE

  ON VARIABLE
    Schema . . . ________  >
    Name . . . . __________________  >

  TO . . . . . . ________________________________________________________________
 ______________________________________________________________________________  >
```

Figure 510. Grant Variable Privileges panel (ADBPGGV)

- In the **ALL**, **READ**, or **WRITE** fields, specify Y or G to indicate the type of authorization that you want to grant. (G indicates the WITH GRANT option, which allows the user to grant privileges to other users.)
- In the **ON VARIABLE** section, specify the qualified name of the global variable for which you are granting privileges.
- In the **TO** field, specify the user ID or IDs or role to which you want to grant access. If you specify more than one user ID, separate each one by a comma. If you specify a role, specify the ROLE keyword and a defined role name, such as ROLE *groupadm*.

Related tasks

“Requesting revoke impact reports in batch” on page 517
A *revoke impact report* helps you determine how the authorizations and database objects will be affected by revoking an authorization before you actually revoke it. You can request these reports on the **Revoke *object* Privileges** panels. You can also request these reports by using a batch job.

Related reference

GRANT (variable privileges) (Db2 12 for z/OS)
REVOKE (variable privileges) (Db2 12 for z/OS)

Revoking global variable authorizations

You can revoke the authority that users have to grant privileges to global variables. You can also revoke the privileges that users have to use global variables. Db2 Admin Tool guides you through the process without requiring you to know the syntax of the REVOKE SQL statements.

About this task

Restriction: You cannot revoke a privilege from a global variable if any of the following conditions exist:

- A function that is owned by the revokee references (READ or WRITE privilege) the specified global variable.
- A view that is owned by the revokee references (READ or WRITE privilege) the specified global variable.
- A trigger that is owned by the revokee references (READ or WRITE privilege) the specified global variable.
- A procedure that is owned by the revokee references (READ or WRITE privilege) the specified global variable.

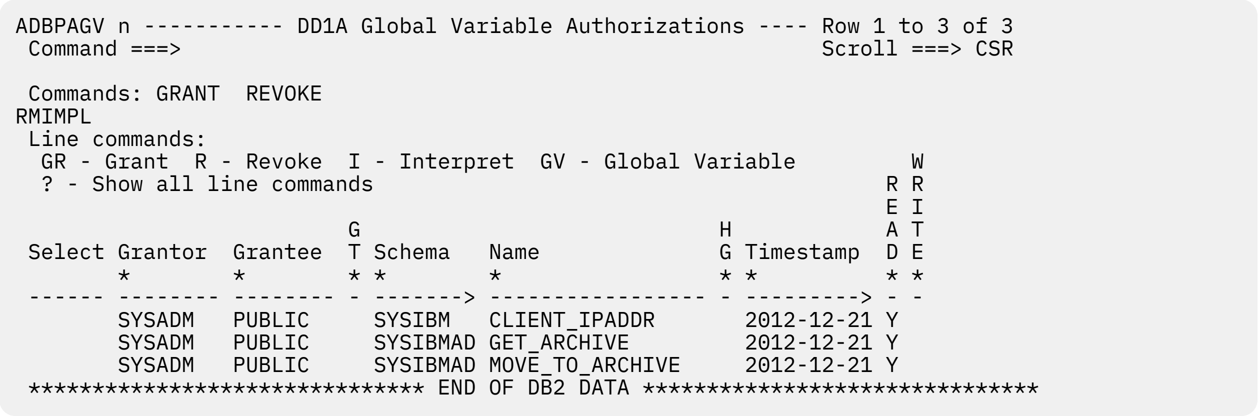
Procedure

To revoke global variable authorizations:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option 1, and press Enter.
2. On the **System Catalog (ADB21)** panel, specify the AO command, and press Enter.

   The authorization options are displayed on the **System Catalog (ADB21)** panel.
3. Specify option GVA, and press Enter.

   The **Global Variable Authorizations (ADBPAGV)** panel is displayed, as shown in the following figure.

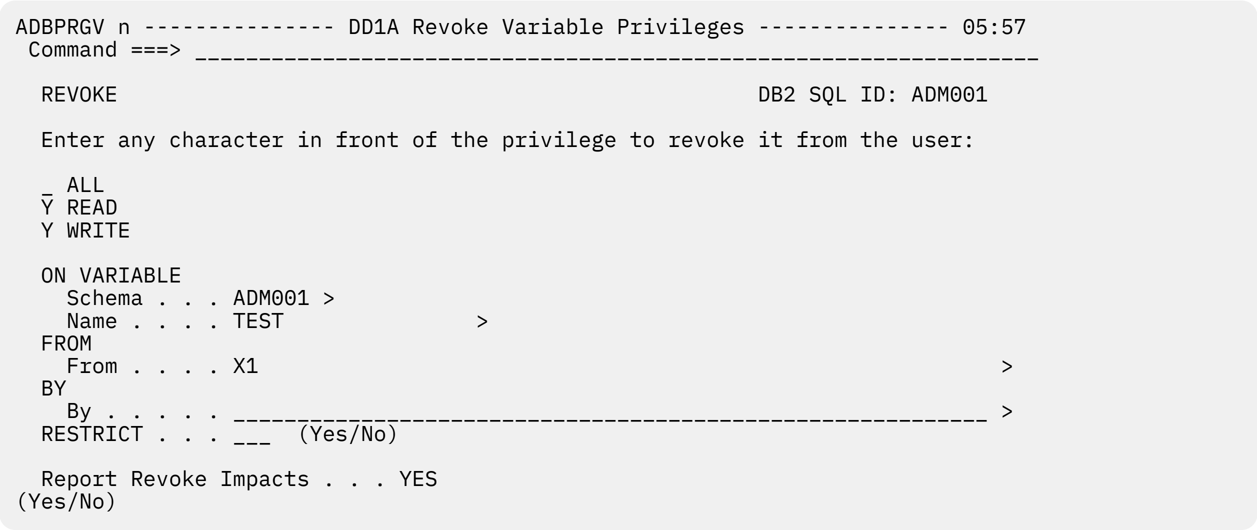


```
ADBPAGV n ----------- DD1A Global Variable Authorizations ---- Row 1 to 3 of 3
 Command ===>                                                  Scroll ===> CSR

 Commands: GRANT  REVOKE
RMIMPL
 Line commands:
  GR - Grant  R - Revoke  I - Interpret  GV - Global Variable         W
  ? - Show all line commands                                        R R
                                                                    E I
                          G                            H            A T
 Select Grantor  Grantee  T Schema   Name             G Timestamp  D E
        *        *        * *        *                 * *          * *
 ------ -------- -------- - -------> ----------------- - ---------> - -
        SYSADM   PUBLIC     SYSIBM   CLIENT_IPADDR       2012-12-21 Y
        SYSADM   PUBLIC     SYSIBMAD GET_ARCHIVE         2012-12-21 Y
        SYSADM   PUBLIC     SYSIBMAD MOVE_TO_ARCHIVE     2012-12-21 Y
 ******************************* END OF DB2 DATA *******************************
```

*Figure 511. **Global Variable Authorizations (ADBPAGV)** panel*

4. Issue the R line command against the global variable whose authorization you want to revoke, and press Enter.

   The **Revoke Variable Privileges (ADBPRGV)** panel is displayed, as shown in the following figure.



```
ADBPRGV n --------------- DD1A Revoke Variable Privileges --------------- 05:57
 Command ===> __________________________________________________________________

  REVOKE                                                  DB2 SQL ID: ADM001

  Enter any character in front of the privilege to revoke it from the user:

  _ ALL
  Y READ
  Y WRITE

  ON VARIABLE
    Schema . . . ADM001 >
    Name . . . . TEST               >
  FROM
    From . . . . X1                                                          >
  BY
    By . . . . . ___________________________________________________________ >
  RESTRICT . . . ___  (Yes/No)

  Report Revoke Impacts . . . YES
(Yes/No)
```

Figure 512. ***Revoke Variable Privileges (ADBPRGV)*** *panel*

5. Specify the privilege that you want to revoke and the **FROM**, **BY**, and **RESTRICT** clause information.

   For more information about these clauses, see REVOKE (variable privileges) (Db2 12 for z/OS).

6. Optional: Review the revoke impact report:

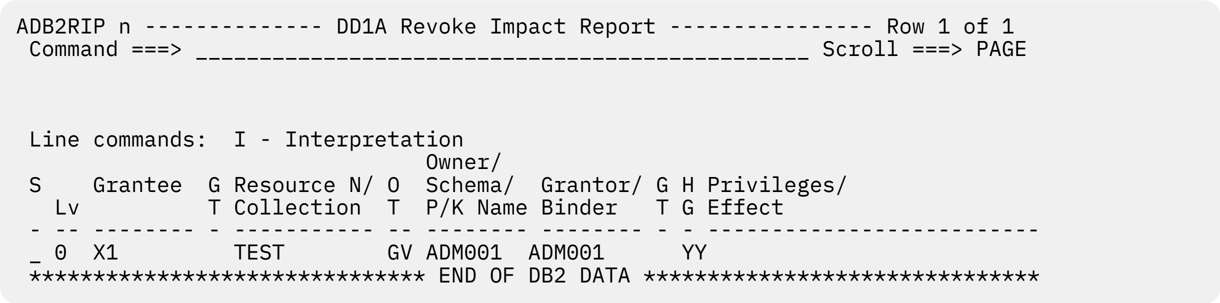
   This report helps you determine how the authorizations and database objects will be affected by revoking an authorization before you actually revoke it.

   a) In the **Report Revoke Impacts** field, specify Yes, and press Enter.

      If the following message is displayed, your user ID does not have the authority to execute the REVOKE statement:

      ```
      Revoker does not have SYSADM/SYSCTRL/SECADM/ACCESSCTRL
      ```

      Otherwise, the **Revoke Impact Report (ADB2RIP)** panel is displayed, as shown in the following figure.



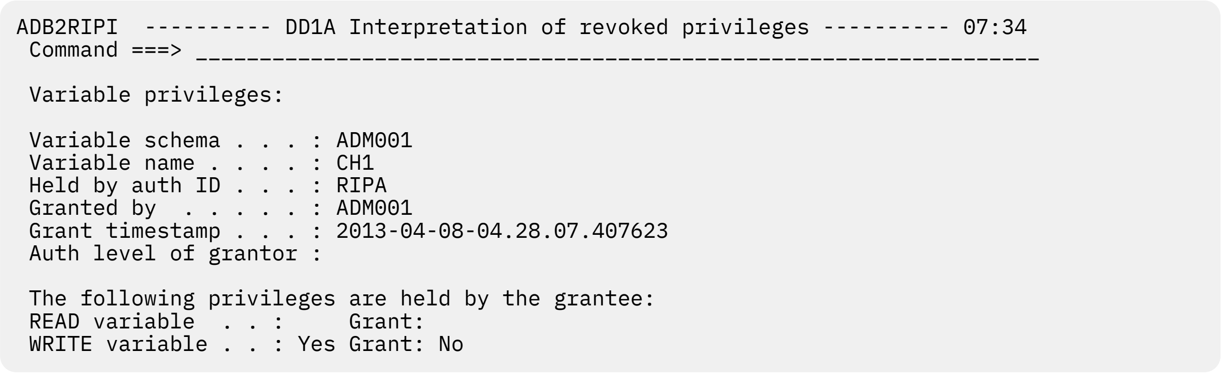
```
ADB2RIP n -------------- DD1A Revoke Impact Report ---------------- Row 1 of 1
 Command ===> ________________________________________________ Scroll ===> PAGE



 Line commands:  I - Interpretation
                                Owner/
 S    Grantee  G Resource N/ O  Schema/  Grantor/ G H Privileges/
   Lv          T Collection  T  P/K Name Binder   T G Effect
 - -- -------- - ----------- -- -------- -------- - - ---------------------------
 _ 0  X1         TEST        GV ADM001  ADM001      YY
 ******************************* END OF DB2 DATA *******************************
```

Figure 513. Revoke Impact report (ADB2RIP)

      For information about any of these columns and their values, see the online help (PF1).

   b) Issue the I line command next to the global variable for which you want to display interpretation information, and press Enter.

      The **Interpretation of revoked privileges (ADB2RIPI)** panel is displayed, as shown in the following figure.



```
ADB2RIPI  ---------- DD1A Interpretation of revoked privileges ---------- 07:34
 Command ===> __________________________________________________________________

 Variable privileges:

 Variable schema . . . : ADM001
 Variable name . . . . : CH1
 Held by auth ID . . . : RIPA
 Granted by  . . . . . : ADM001
 Grant timestamp . . . : 2013-04-08-04.28.07.407623
 Auth level of grantor :

 The following privileges are held by the grantee:
 READ variable  . . :     Grant:
 WRITE variable . . : Yes Grant: No
```

*Figure 514. **Interpretation of revoked privileges (ADB2RIPI)** panel*

c) Exit back to the **Revoke Variable Privileges (ADBPRGV)** panel

7. In the **Report Revoke Impacts** field, specify No, and press Enter.

   A Change Management prompt is displayed that shows you the SQL REVOKE statement.

Revoking all authorizations from a user

You can revoke all of the directly held or explicitly granted authorizations from a user.

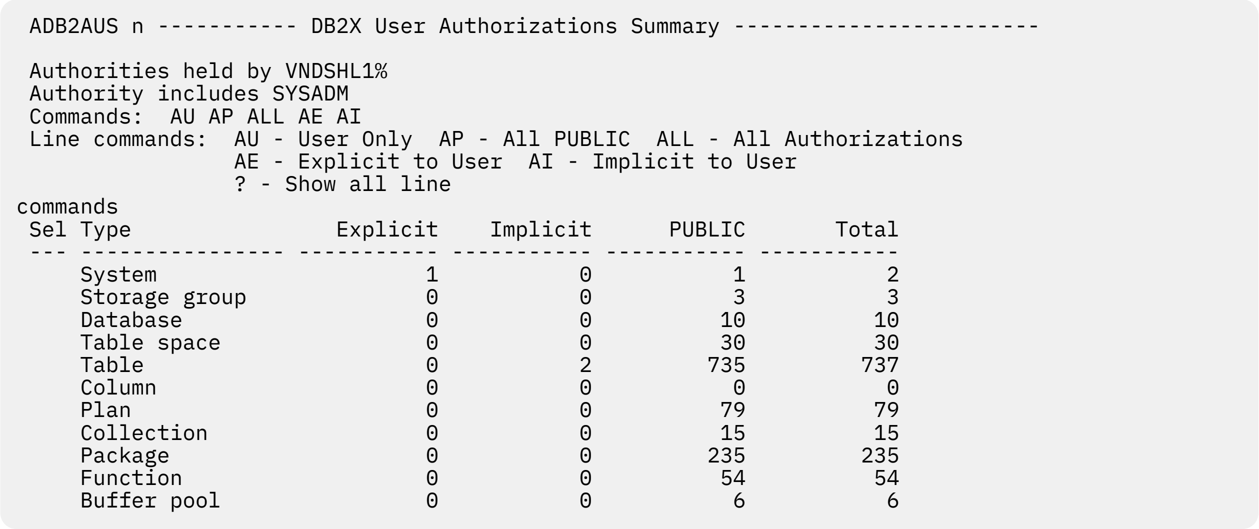
About this task

To revoke the authorizations from a user:

Procedure

1. On the Db2 Admin Tool **System Catalog** panel, type the two-character AO object option in the **Option** field and press Enter.
2. Type the two-character UA authorization option in the **Option** field and specify the name of the user or users from whom to revoke authorizations in the **Grantee** field at the bottom of the panel. Press Enter.

   The **User Authorizations Summary** panel, as shown in the following figure, is displayed.



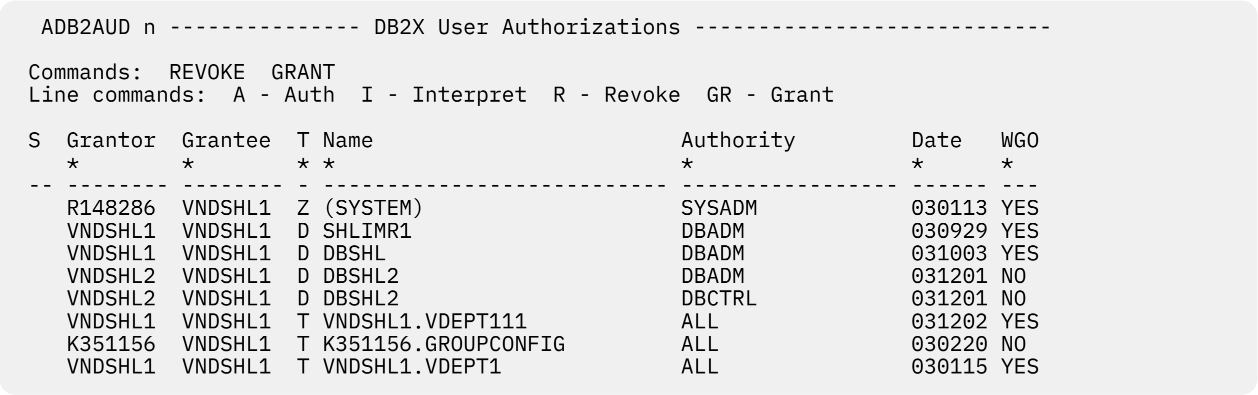
```
 ADB2AUS n ----------- DB2X User Authorizations Summary -----------------------

 Authorities held by VNDSHL1%
 Authority includes SYSADM
 Commands:  AU AP ALL AE AI
 Line commands:  AU - User Only  AP - All PUBLIC  ALL - All Authorizations
                 AE - Explicit to User  AI - Implicit to User
                 ? - Show all line
commands
 Sel Type               Explicit    Implicit      PUBLIC       Total
 --- ---------------- ----------- ----------- ----------- -----------
     System                     1           0           1           2
     Storage group              0           0           3           3
     Database                   0           0          10          10
     Table space                0           0          30          30
     Table                      0           2         735         737
     Column                     0           0           0           0
     Plan                       0           0          79          79
     Collection                 0           0          15          15
     Package                    0           0         235         235
     Function                   0           0          54          54
     Buffer pool                0           0           6           6
```

Figure 515. User Authorizations Summary panel (ADB2AUS)

3. Issue the AU or AE command to display the authorizations that are held by the grantees that you specified.

   AU shows the authorizations that the specified grantees hold directly, and AE shows the authorizations that the specified grantees were granted explicitly.

   The **User Authorizations** panel, as shown in the following figure, is displayed.

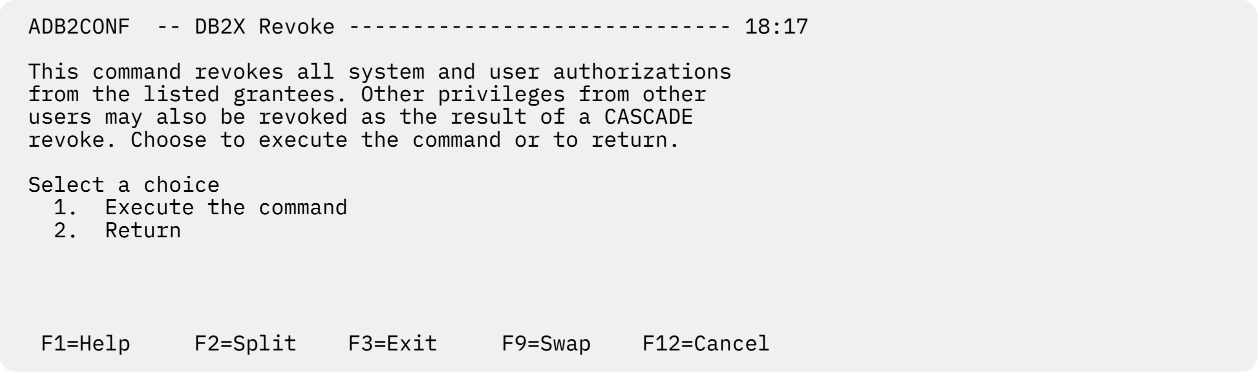


```
 ADB2AUD n --------------- DB2X User Authorizations ----------------------------

Commands:  REVOKE  GRANT
Line commands:  A - Auth  I - Interpret  R - Revoke  GR - Grant

S  Grantor  Grantee  T Name                         Authority         Date   WGO
   *        *        * *                            *                 *      *
-- -------- -------- - -------------------------- ----------------- ------ ---
   R148286  VNDSHL1  Z (SYSTEM)                     SYSADM            030113 YES
   VNDSHL1  VNDSHL1  D SHLIMR1                      DBADM             030929 YES
   VNDSHL1  VNDSHL1  D DBSHL                        DBADM             031003 YES
   VNDSHL2  VNDSHL1  D DBSHL2                       DBADM             031201 NO
   VNDSHL2  VNDSHL1  D DBSHL2                       DBCTRL            031201 NO
   VNDSHL1  VNDSHL1  T VNDSHL1.VDEPT111             ALL               031202 YES
   K351156  VNDSHL1  T K351156.GROUPCONFIG          ALL               030220 NO
   VNDSHL1  VNDSHL1  T VNDSHL1.VDEPT1               ALL               030115 YES
```

Figure 516. User Authorizations panel (ADB2AUD)

4. Issue the REVOKE primary command to revoke all of the listed system and user authorities from the listed grantees.

   The **Revoke** panel, as shown in the following figure, is displayed to remind you of the significant impact that executing the command can have and to have you confirm whether you really want to execute it.



```
ADB2CONF  -- DB2X Revoke ------------------------------ 18:17

This command revokes all system and user authorizations
from the listed grantees. Other privileges from other
users may also be revoked as the result of a CASCADE
revoke. Choose to execute the command or to return.

Select a choice
  1.  Execute the command
  2.  Return



 F1=Help     F2=Split    F3=Exit     F9=Swap    F12=Cancel
```

Figure 517. Revoke panel (ADB2CONF)

5. Specify option 1 to execute the REVOKE command.

   The SQL is generated and executed if the total size of the generated SQL is less than 32K (approximately 60 REVOKE statements). Otherwise, the **Statement Execution Prompt** panel is displayed, and you can choose to create a batch job with the statements or add the statements to a work statement list (WSL).

Granting a set of authorizations to a user

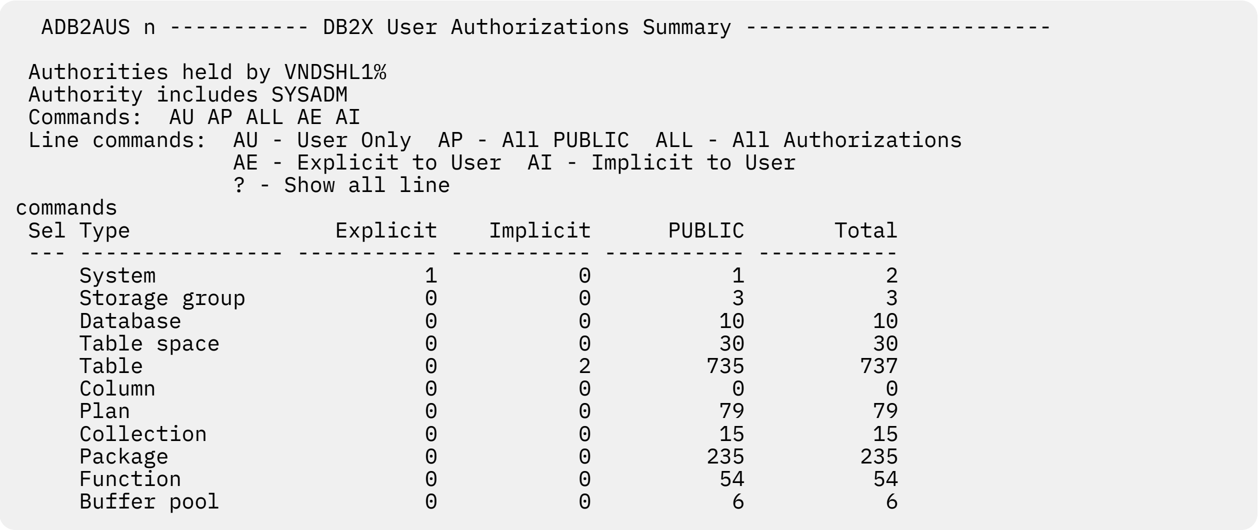
When managing authorizations, you might want to give all the authorizations that are held by one user, either those held directly or those granted explicitly, to another user or a list of users.

Procedure

To grant all the authorizations that are held by one user to another user:

1. On the Db2 Admin Tool **System Catalog** panel, type the two-character AO object option in the **Option** field and press Enter.
2. Type the two-character UA authorization option in the **Option** field and specify the name of the user from whom to copy authorizations in the **Grantee** field at the bottom of the panel. Press Enter.

   The **User Authorizations Summary** panel, as shown in the following figure, is displayed.



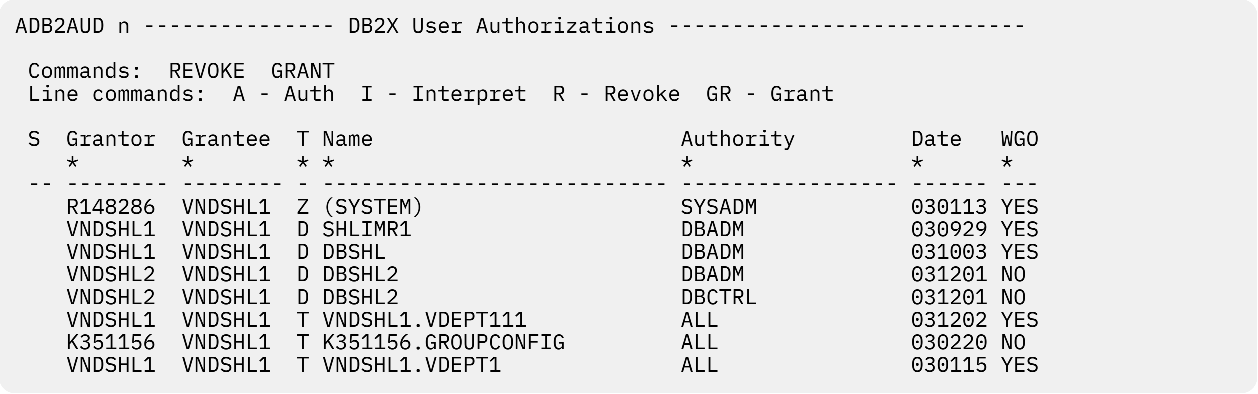
```
  ADB2AUS n ----------- DB2X User Authorizations Summary -----------------------

 Authorities held by VNDSHL1%
 Authority includes SYSADM
 Commands:  AU AP ALL AE AI
 Line commands:  AU - User Only  AP - All PUBLIC  ALL - All Authorizations
                 AE - Explicit to User  AI - Implicit to User
                 ? - Show all line
commands
 Sel Type                Explicit    Implicit      PUBLIC       Total
 --- ---------------- ----------- ----------- ----------- -----------
     System                     1           0           1           2
     Storage group              0           0           3           3
     Database                   0           0          10          10
     Table space                0           0          30          30
     Table                      0           2         735         737
     Column                     0           0           0           0
     Plan                       0           0          79          79
     Collection                 0           0          15          15
     Package                    0           0         235         235
     Function                   0           0          54          54
     Buffer pool                0           0           6           6
```

Figure 518. User Authorizations Summary panel (ADB2AUS)

3. Issue the AU or AE command to display the authorizations that are held by the grantee that you specified.

   AU shows the authorizations that the specified grantee holds directly, and AE shows the authorizations that the specified grantee was granted explicitly.

   The **User Authorizations** panel, as shown in the following figure, is displayed.

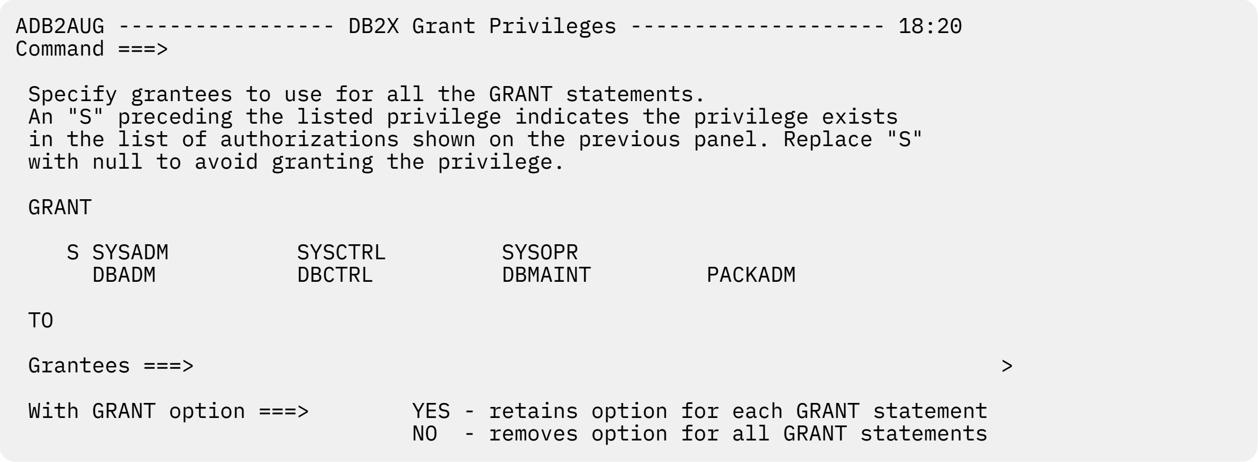


```
ADB2AUD n --------------- DB2X User Authorizations ----------------------------

 Commands:  REVOKE  GRANT
 Line commands:  A - Auth  I - Interpret  R - Revoke  GR - Grant

 S  Grantor  Grantee  T Name                          Authority         Date   WGO
    *        *        * *                             *                 *      *
 -- -------- -------- - --------------------------- ---------------- ------ ---
    R148286  VNDSHL1  Z (SYSTEM)                     SYSADM           030113 YES
    VNDSHL1  VNDSHL1  D SHLIMR1                      DBADM            030929 YES
    VNDSHL1  VNDSHL1  D DBSHL                        DBADM            031003 YES
    VNDSHL2  VNDSHL1  D DBSHL2                       DBADM            031201 NO
    VNDSHL2  VNDSHL1  D DBSHL2                       DBCTRL           031201 NO
    VNDSHL1  VNDSHL1  T VNDSHL1.VDEPT111             ALL              031202 YES
    K351156  VNDSHL1  T K351156.GROUPCONFIG          ALL              030220 NO
    VNDSHL1  VNDSHL1  T VNDSHL1.VDEPT1               ALL              030115 YES
```

Figure 519. User Authorizations panel (ADB2AUD)

4. Issue the GRANT primary command.

   The **Grant Privileges** panel, as shown in the following figure, is displayed.



```
ADB2AUG ----------------- DB2X Grant Privileges -------------------- 18:20
Command ===>

 Specify grantees to use for all the GRANT statements.
 An "S" preceding the listed privilege indicates the privilege exists
 in the list of authorizations shown on the previous panel. Replace "S"
 with null to avoid granting the privilege.

 GRANT

    S SYSADM          SYSCTRL         SYSOPR
      DBADM           DBCTRL          DBMAINT         PACKADM

 TO

 Grantees ===>                                                      >

 With GRANT option ===>        YES - retains option for each GRANT statement
                               NO  - removes option for all GRANT statements
```

Figure 520. Grant Privileges panel (ADB2AUG)

5. Specify the users to whom you would like to grant authorizations in the Grantees field.

   The SQL is generated and executed if the size of the generated SQL is less than 32K. Otherwise, the **Statement Execution Prompt** panel is displayed, and you can choose to create a batch job with the statements or add the statements to a work statement list (WSL).

Displaying buffer pool status

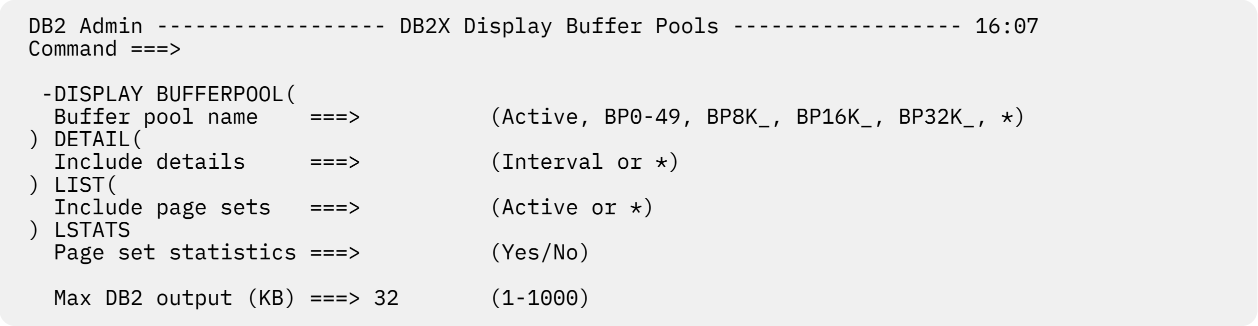
You can display the current status of one or more active or inactive buffer pools.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option BD, and press Enter.

   The **Display Buffer Pools (ADB2ZBD)** panel is displayed, as shown in the following figure.



```
DB2 Admin ------------------ DB2X Display Buffer Pools ------------------ 16:07
Command ===>

 -DISPLAY BUFFERPOOL(
  Buffer pool name    ===>          (Active, BP0-49, BP8K_, BP16K_, BP32K_, *)
) DETAIL(
  Include details     ===>          (Interval or *)
) LIST(
  Include page sets   ===>          (Active or *)
) LSTATS
  Page set statistics ===>          (Yes/No)

  Max DB2 output (KB) ===> 32       (1-1000)
```

*Figure 521. **Display Buffer Pools (ADB2ZBD)** panel*

3. Specify the appropriate keywords and parameters on the panel.

   Db2 Admin Tool issues the Db2 DISPLAY BUFFERPOOL command. The information that Db2 Admin Tool returns to you from the command is in ISPF browse format.

Altering buffer pools

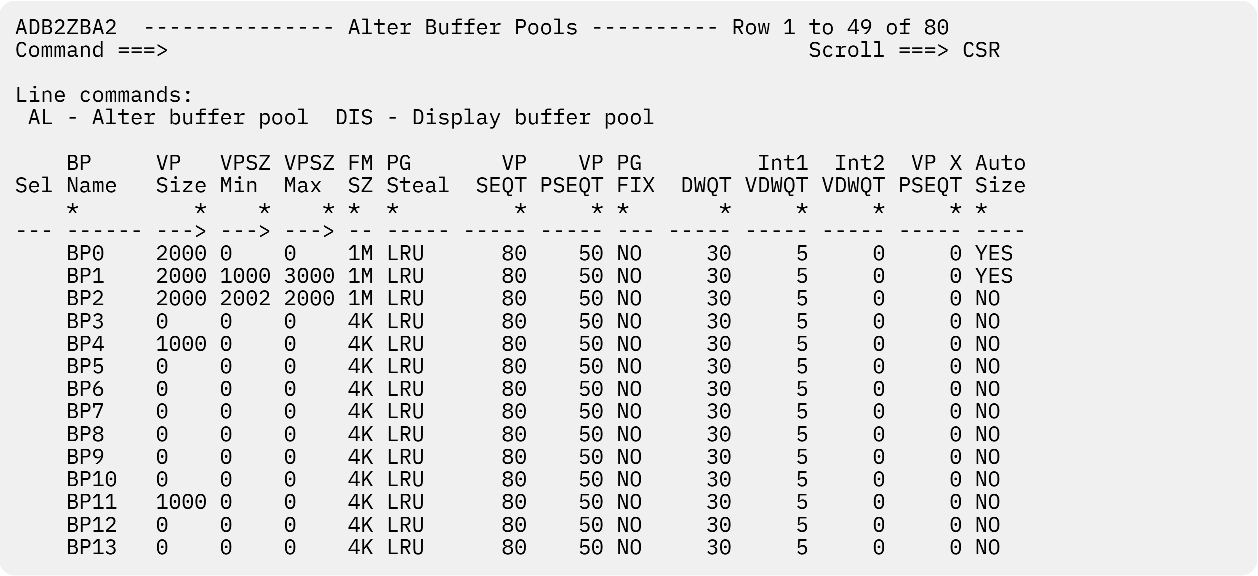
You can alter the attributes of active or inactive buffer pools.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option BA, and press Enter.

   The **Alter Buffer Pools (ADB2ZBA2)** panel is displayed, as shown in the following figure.



```
ADB2ZBA2  --------------- Alter Buffer Pools ---------- Row 1 to 49 of 80
Command ===>                                                  Scroll ===> CSR

Line commands:
 AL - Alter buffer pool  DIS - Display buffer pool

    BP     VP   VPSZ VPSZ FM PG       VP    VP PG         Int1  Int2  VP X Auto
Sel Name   Size Min  Max  SZ Steal  SEQT PSEQT FIX  DWQT VDWQT VDWQT PSEQT Size
    *         *    *    * *  *         *     * *       *     *     *     * *
--- ------ ---> ---> ---> -- ----- ----- ----- --- ----- ----- ----- ----- ----
    BP0    2000 0    0    1M LRU      80    50 NO     30     5     0     0 YES
    BP1    2000 1000 3000 1M LRU      80    50 NO     30     5     0     0 YES
    BP2    2000 2002 2000 1M LRU      80    50 NO     30     5     0     0 NO
    BP3    0    0    0    4K LRU      80    50 NO     30     5     0     0 NO
    BP4    1000 0    0    4K LRU      80    50 NO     30     5     0     0 NO
    BP5    0    0    0    4K LRU      80    50 NO     30     5     0     0 NO
    BP6    0    0    0    4K LRU      80    50 NO     30     5     0     0 NO
    BP7    0    0    0    4K LRU      80    50 NO     30     5     0     0 NO
    BP8    0    0    0    4K LRU      80    50 NO     30     5     0     0 NO
    BP9    0    0    0    4K LRU      80    50 NO     30     5     0     0 NO
    BP10   0    0    0    4K LRU      80    50 NO     30     5     0     0 NO
    BP11   1000 0    0    4K LRU      80    50 NO     30     5     0     0 NO
    BP12   0    0    0    4K LRU      80    50 NO     30     5     0     0 NO
    BP13   0    0    0    4K LRU      80    50 NO     30     5     0     0 NO
```

*Figure 522. **Alter Buffer Pools (ADB2ZBA2)** panel*

The following columns are available on this panel:

Sel
Input field where you enter one of the line commands listed on the panel.

BP NAME
Buffer pool name.

VP SIZE
Virtual buffer pool size.

VPSZ MIN
The minimum size for the buffer pool.

VPSZ MAX
The maximum size for the buffer pool.

FM SZ
The frame size for the buffer pool.

HP SIZE
Hiperpool size.

CAST OUT
Hiperspace* CASTOUT value.

VP SEQT
Virtual sequential steal threshold.

VP PSEQT
Virtual parallel sequential threshold.

HP SEQT
Hiperpool sequential steal threshold.

DWQT
Deferred write threshold.

VDWQT
Vertical deferred write threshold.

VP X PSEQT
Assisting virtual parallel sequential threshold.

Auto Size
Specifies whether the buffer pool adjustment is turned on or off.

NO
: Specifies that the buffer pool does not use Workload Manager (WLM) services for automatic buffer pool sizing adjustment. This is the default.

YES
: Specifies that the buffer pool uses WLM services, if available, to automatically adjust the size of the buffer pool based on dynamic monitoring of the workload goals and the available storage on the system.

3. Issue one of the following line commands:

   - AL to alter a buffer pool. When you press Enter, Db2 Admin Tool issues the -ALTER BUFFERPOOL command.
   - DIS to display buffer pool. When you press Enter, Db2 Admin Tool issues the -DISPLAY BUFFERPOOL command.

   The information Db2 Admin Tool returns to you from the commands is in ISPF browse format.

Displaying buffer pool hit ratios

You can name the buffer pools for which buffer pool hit ratios should be displayed.

About this task

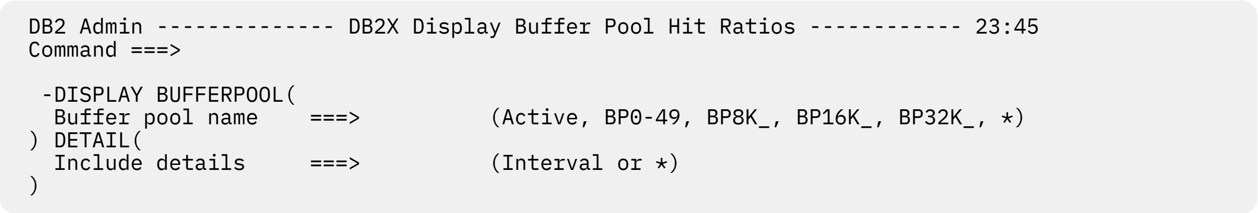
The hit ratio is calculated as the number of hits in the buffer pool divided by the number of GETPAGES.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option BH, and press Enter.

   The **Display Buffer Pool Hit Ratios (ADB2ZBH)** panel is displayed, as shown in the following figure.



```
DB2 Admin -------------- DB2X Display Buffer Pool Hit Ratios ------------ 23:45
Command ===>

 -DISPLAY BUFFERPOOL(
  Buffer pool name    ===>          (Active, BP0-49, BP8K_, BP16K_, BP32K_, *)
) DETAIL(
  Include details     ===>          (Interval or *)
)
```

*Figure 523. **Display Buffer Pool Hit Ratios (ADB2ZBH)** panel*

3. Enter the name of a buffer pool.

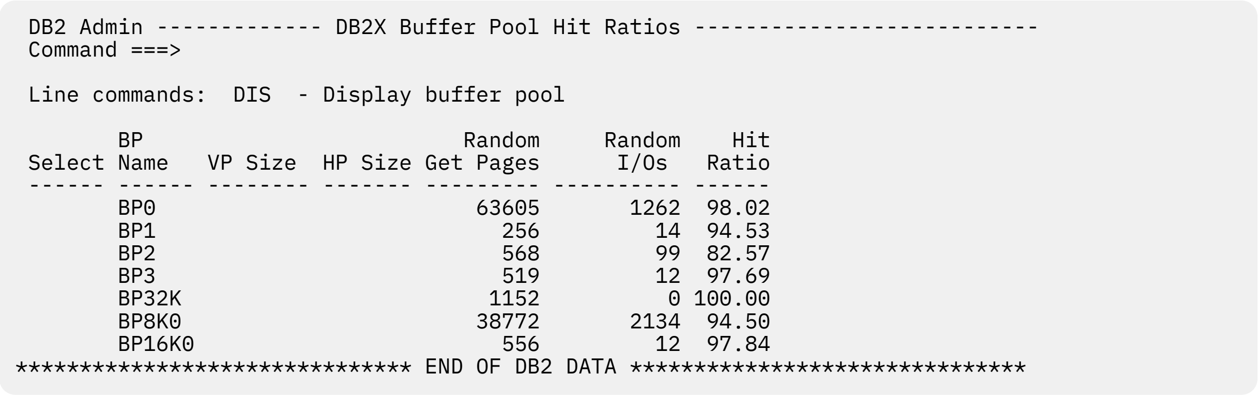
   The following values are valid:

   Active
   : All active buffer pools.

   BP0–BP49, BP8K_, BP16K_, BP32K_
   : Select a specific buffer pool name from the valid values available.

   *****
   : All buffer pools.
4. Specify the interval for which information should be displayed; the interval can be either since the buffer pool was created (*) or since the last display (interval).
5. Press Enter.

   Db2 Admin Tool issues the Db2 DISPLAY BUFFERPOOL command to generate the **Buffer Pool Hit Ratios (ADB2ZBH2)** panel panel, as shown in the following figure.



```
 DB2 Admin ------------- DB2X Buffer Pool Hit Ratios --------------------------
 Command ===>

 Line commands:  DIS  - Display buffer pool

        BP                          Random     Random    Hit
 Select Name   VP Size  HP Size Get Pages      I/Os   Ratio
 ------ ------ -------- ------- --------- ---------- ------
        BP0                         63605       1262  98.02
        BP1                           256         14  94.53
        BP2                           568         99  82.57
        BP3                           519         12  97.69
        BP32K                        1152          0 100.00
        BP8K0                       38772       2134  94.50
        BP16K0                        556         12  97.84
******************************* END OF DB2 DATA *******************************
```

*Figure 524. **Buffer Pool Hit Ratios (ADB2ZBH2)** panel*

The following columns are available on this panel:

Select
Input field where you list one of the line commands listed on the panel.

BP NAME
Name of the buffer pool.

VP SIZE
Size of the virtual buffer pool.

HP SIZE
Size of the hiperpool.

RANDOM GET PAGES
Number of random GETPAGES (RGP).

RANDOM I/Os
Number of random I/Os (RIO).

HIT RATIO
Buffer pool hit ratio, which is calculated as follows:

```
     100 * (RGP - RIO) / RGP
```

Viewing group buffer pools

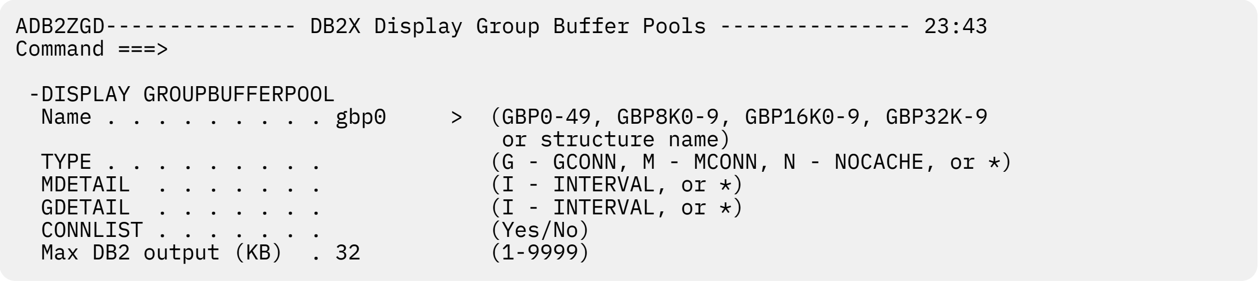
You can view buffer pools that are in Db2 data sharing.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option GD, and press Enter.

   The **Display Group Buffer Pool (ADB2ZGD)** panel is displayed, as shown in the following figure.



```
ADB2ZGD--------------- DB2X Display Group Buffer Pools --------------- 23:43
Command ===>

 -DISPLAY GROUPBUFFERPOOL
  Name . . . . . . . . . gbp0     >  (GBP0-49, GBP8K0-9, GBP16K0-9, GBP32K-9
                                      or structure name)
  TYPE . . . . . . . . .             (G - GCONN, M - MCONN, N - NOCACHE, or *)
  MDETAIL  . . . . . . .             (I - INTERVAL, or *)
  GDETAIL  . . . . . . .             (I - INTERVAL, or *)
  CONNLIST . . . . . . .             (Yes/No)
  Max DB2 output (KB)  . 32          (1-9999)
```

*Figure 525. **Display Group Buffer Pool (ADB2ZGD)** panel*

The following fields are available on this panel:

NAME
Group buffer pool name.

TYPE
Specifies the type of group buffer pools.

GCONN
Group buffer pools that are currently connected to any member of the data sharing group.

MCONN
Group buffer pools that are currently connected to the member to which the command is directed.

NOCACHE
Group buffer pools that have the GBPCACHE attribute set to NO.

MDETAIL
Shows a detailed statistical report that lists the member's activity for each group buffer pool. If a group member has never been actively connected to the group buffer pool, no detail report is shown. The default is interval, which means the report shows incremental statistics.

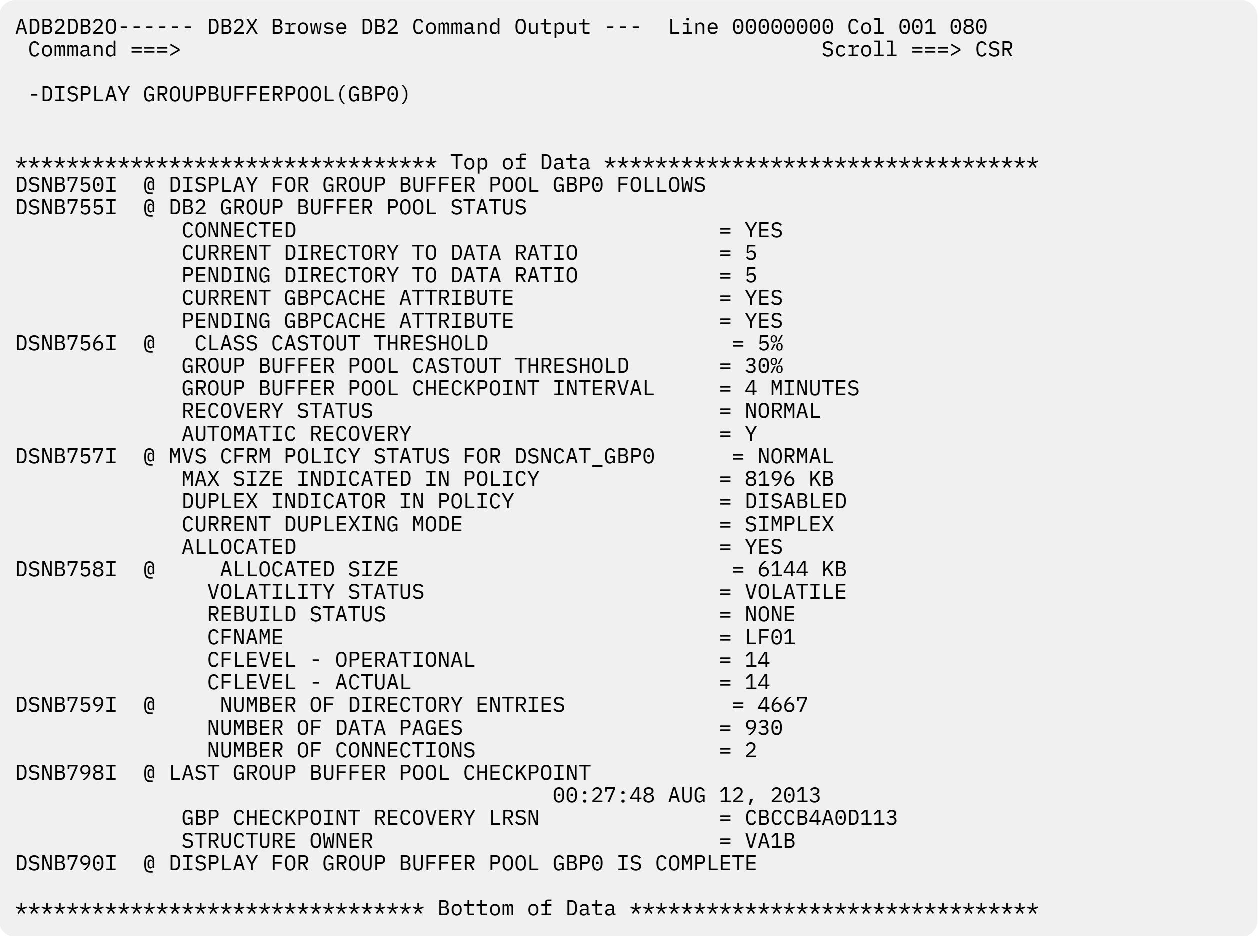
GDETAIL
Shows a detailed statistical report that lists the activity of the entire group for each group buffer pool. If a group member is not actively connected to the group buffer pool, no detail report is shown.

CONNLIST
Specifies whether a connection list report is shown for the specified group buffer pools. The report lists the connection names of the subsystems that are currently connected to the group buffer pools and provides connection status.

Max DB2 output
Specifies the maximum size of ISPF table that stores the report for the group buffer pool.

3. Optional: Press Enter to run the **DISPLAY GROUPBUFFERPOOL** command.

   The **Browse DB2 Command Output (ADB2DB2O)** panel is displayed, as shown in the following figure.



```
ADB2DB2O------ DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> CSR

 -DISPLAY GROUPBUFFERPOOL(GBP0)


********************************* Top of Data **********************************
DSNB750I  @ DISPLAY FOR GROUP BUFFER POOL GBP0 FOLLOWS
DSNB755I  @ DB2 GROUP BUFFER POOL STATUS
             CONNECTED                                 = YES
             CURRENT DIRECTORY TO DATA RATIO           = 5
             PENDING DIRECTORY TO DATA RATIO           = 5
             CURRENT GBPCACHE ATTRIBUTE                = YES
             PENDING GBPCACHE ATTRIBUTE                = YES
DSNB756I  @   CLASS CASTOUT THRESHOLD                   = 5%
             GROUP BUFFER POOL CASTOUT THRESHOLD       = 30%
             GROUP BUFFER POOL CHECKPOINT INTERVAL     = 4 MINUTES
             RECOVERY STATUS                           = NORMAL
             AUTOMATIC RECOVERY                        = Y
DSNB757I  @ MVS CFRM POLICY STATUS FOR DSNCAT_GBP0      = NORMAL
             MAX SIZE INDICATED IN POLICY              = 8196 KB
             DUPLEX INDICATOR IN POLICY                = DISABLED
             CURRENT DUPLEXING MODE                    = SIMPLEX
             ALLOCATED                                 = YES
DSNB758I  @     ALLOCATED SIZE                          = 6144 KB
               VOLATILITY STATUS                       = VOLATILE
               REBUILD STATUS                          = NONE
               CFNAME                                  = LF01
               CFLEVEL - OPERATIONAL                   = 14
               CFLEVEL - ACTUAL                        = 14
DSNB759I  @     NUMBER OF DIRECTORY ENTRIES             = 4667
               NUMBER OF DATA PAGES                    = 930
               NUMBER OF CONNECTIONS                   = 2
DSNB798I  @ LAST GROUP BUFFER POOL CHECKPOINT
                                          00:27:48 AUG 12, 2013
             GBP CHECKPOINT RECOVERY LRSN              = CBCCB4A0D113
             STRUCTURE OWNER                           = VA1B
DSNB790I  @ DISPLAY FOR GROUP BUFFER POOL GBP0 IS COMPLETE

******************************** Bottom of Data ********************************
```

Figure 526. ***Browse DB2 Command Output (ADB2DB2O)*** *panel*

Altering group buffer pools

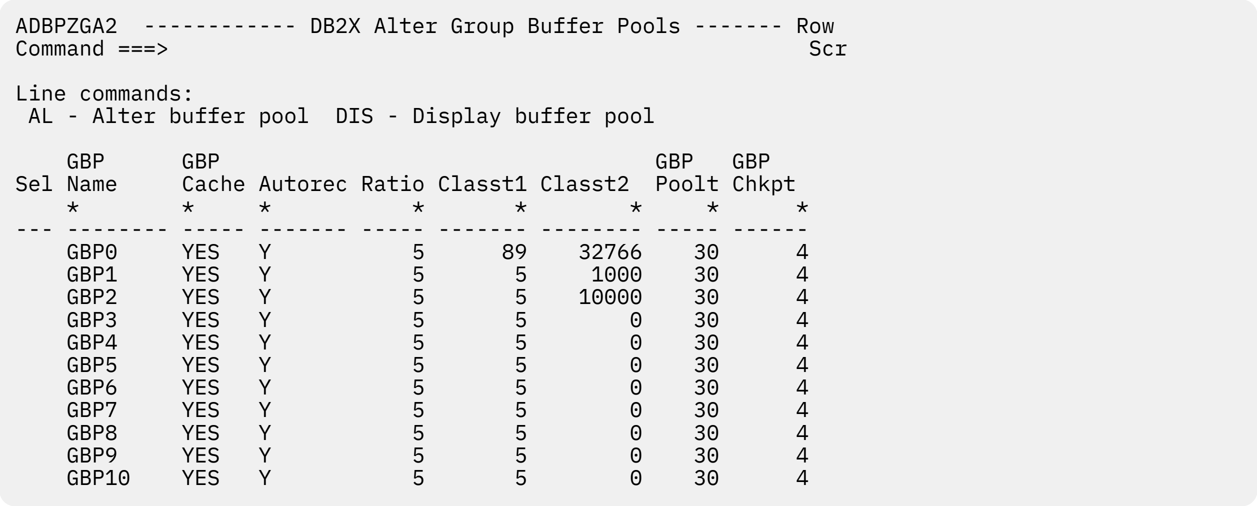
You can alter the information for group buffer pools that are in Db2 data sharing.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option GA, and press Enter.

   The **Alter Group Buffer Pools (ADBPZGA2)** panel is displayed, as shown in the following figure.



```
ADBPZGA2  ------------ DB2X Alter Group Buffer Pools ------- Row
Command ===>                                                  Scr

Line commands:
 AL - Alter buffer pool  DIS - Display buffer pool

    GBP      GBP                                  GBP   GBP
Sel Name     Cache Autorec Ratio Classt1 Classt2  Poolt Chkpt
    *        *     *           *       *        *     *      *
--- -------- ----- ------- ----- ------- -------- ----- ------
    GBP0     YES   Y           5      89    32766    30      4
    GBP1     YES   Y           5       5     1000    30      4
    GBP2     YES   Y           5       5    10000    30      4
    GBP3     YES   Y           5       5        0    30      4
    GBP4     YES   Y           5       5        0    30      4
    GBP5     YES   Y           5       5        0    30      4
    GBP6     YES   Y           5       5        0    30      4
    GBP7     YES   Y           5       5        0    30      4
    GBP8     YES   Y           5       5        0    30      4
    GBP9     YES   Y           5       5        0    30      4
    GBP10    YES   Y           5       5        0    30      4
```

Figure 527. ***Alter Group Buffer Pools (ADBPZGA2)*** *panel*

The following columns are available on this panel:

Sel
Input field where you enter one of the line commands that are listed on the panel.

GBPName
Group buffer pool name.

GBPCache
Shows the pending group buffer pool cache attribute. The value **Yes** indicates that the group buffer pool is used for both caching and cross-invalidation.

Autorec
Indicates whether automatic recovery is specified for the group buffer pool.

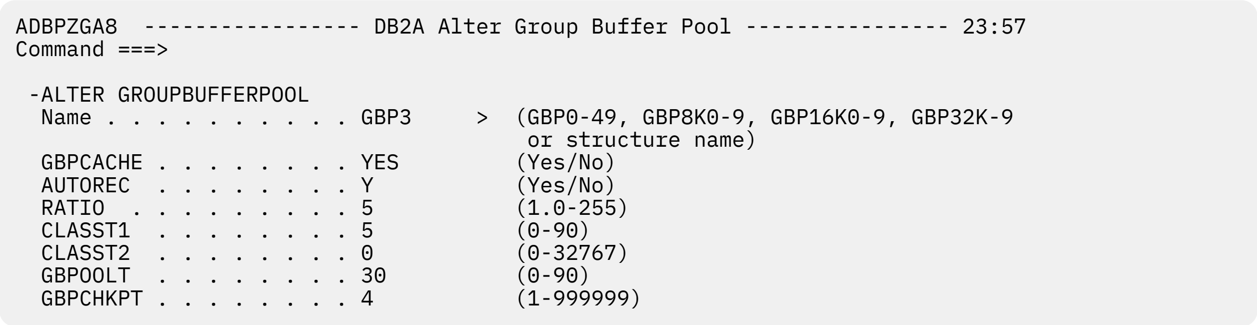
Classt1
Shows a percentage that indicates the degree to which data entries fill the data pages in the group buffer pool.

GBPPoolt
Displays the castout threshold for a group buffer pool. When the threshold is met, the data in the group buffer pool is cast out to disk.

GBPChkpt
Shows the checkpoint interval for a group buffer pool.

3. Choose one group buffer pool, and specify the line command AL in the **Sel** column.

   The **Alter Group Buffer Pools (ADBPZGA8)** panel is displayed, as shown in the following figure.



```
ADBPZGA8  ----------------- DB2A Alter Group Buffer Pool ---------------- 23:57
Command ===>

 -ALTER GROUPBUFFERPOOL
  Name . . . . . . . . . . GBP3     >  (GBP0-49, GBP8K0-9, GBP16K0-9, GBP32K-9
                                        or structure name)
  GBPCACHE . . . . . . . . YES         (Yes/No)
  AUTOREC  . . . . . . . . Y           (Yes/No)
  RATIO  . . . . . . . . . 5           (1.0-255)
  CLASST1  . . . . . . . . 5           (0-90)
  CLASST2  . . . . . . . . 0           (0-32767)
  GBPOOLT  . . . . . . . . 30          (0-90)
  GBPCHKPT . . . . . . . . 4           (1-999999)
```

Figure 528. ***Alter Group Buffer Pools (ADBPZGA8)*** *panel*

4. Optional: Change the group buffer pool parameters.

   The following fields are available on this panel for you to alter:

Name
Group buffer pool name.

GBPCache
Specifies whether group buffer pool is to be used for both caching data and cross-invalidation, or just for cross-invalidation.

Autorec
Specifies whether automatic recovery by Db2 takes place when a structure failure occurs, or when the connectivity to all members of the group buffer pool is lost.

Classt1
A percentage of the number of data entries and can be an integer 0 - 90, inclusive. The default is 5.

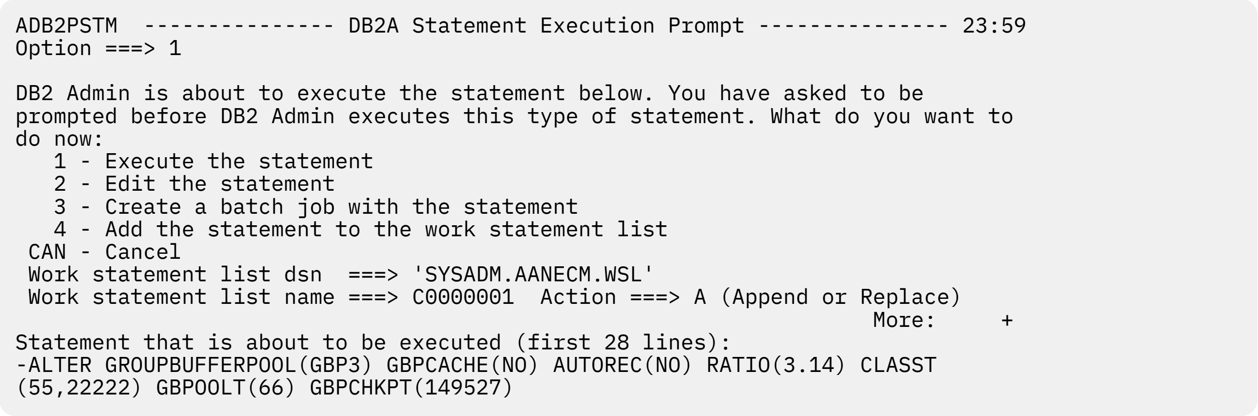
Classt2
An absolute number of pages.

GBPPoolt
The threshold at which data in the group buffer pool is cast out to disk.

GBPChkpt
Changes the time interval, in minutes, between successive checkpoints of the group buffer pool.

5. Press Enter to run the **ALTER GROUPBUFFERPOOL** command.

   The **Statement Execution Prompt (ADB2PSTM)** panel is displayed.



```
ADB2PSTM  --------------- DB2A Statement Execution Prompt --------------- 23:59
Option ===> 1

DB2 Admin is about to execute the statement below. You have asked to be
prompted before DB2 Admin executes this type of statement. What do you want to
do now:
   1 - Execute the statement
   2 - Edit the statement
   3 - Create a batch job with the statement
   4 - Add the statement to the work statement list
 CAN - Cancel
 Work statement list dsn  ===> 'SYSADM.AANECM.WSL'
 Work statement list name ===> C0000001  Action ===> A (Append or Replace)
                                                             More:     +
Statement that is about to be executed (first 28 lines):
-ALTER GROUPBUFFERPOOL(GBP3) GBPCACHE(NO) AUTOREC(NO) RATIO(3.14) CLASST
(55,22222) GBPOOLT(66) GBPCHKPT(149527)
```

*Figure 529. **Statement Execution Prompt (ADB2PSTM)** panel*

6. After the command runs, return to the **Alter Group Buffer Pools (ADBPZGA2)** panel to see the changes that you made.

Displaying archive log information

You can display information about the input archive log.

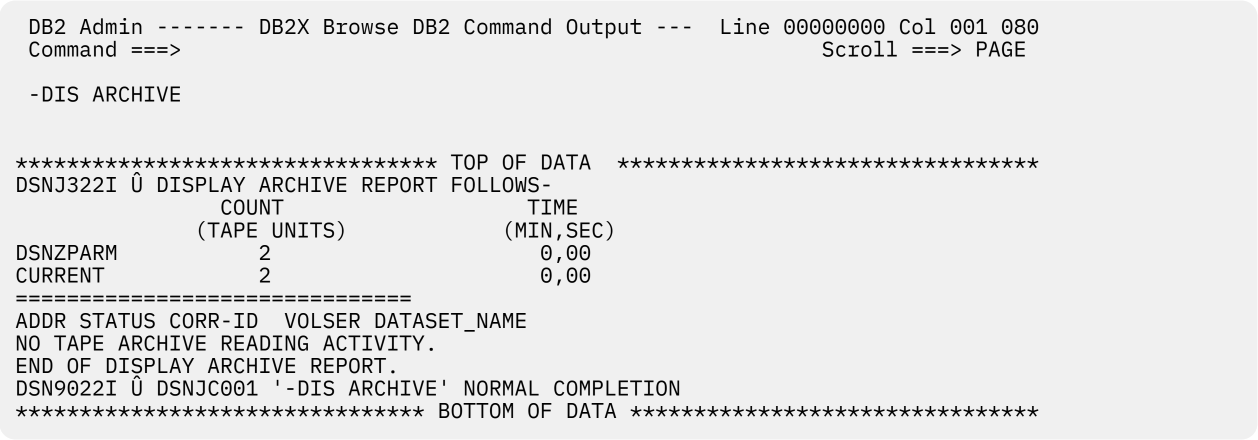
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.

2. Specify option LD, and press Enter.

   The **Display Archive Log (ADB2DB2O)** panel is displayed, as shown in the following figure.



```
 DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

 -DIS ARCHIVE


********************************* TOP OF DATA  *********************************
DSNJ322I Û DISPLAY ARCHIVE REPORT FOLLOWS-
                COUNT                   TIME
              (TAPE UNITS)            (MIN,SEC)
DSNZPARM           2                     0,00
CURRENT            2                     0,00
===============================
ADDR STATUS CORR-ID  VOLSER DATASET_NAME
NO TAPE ARCHIVE READING ACTIVITY.
END OF DISPLAY ARCHIVE REPORT.
DSN9022I Û DSNJC001 '-DIS ARCHIVE' NORMAL COMPLETION
******************************** BOTTOM OF DATA ********************************
```

*Figure 530. **Display Archive Log (ADB2DB2O)** panel*

Db2 Admin Tool generates this panel by issuing the -DISPLAY ARCHIVE command.

Setting archive log parameters

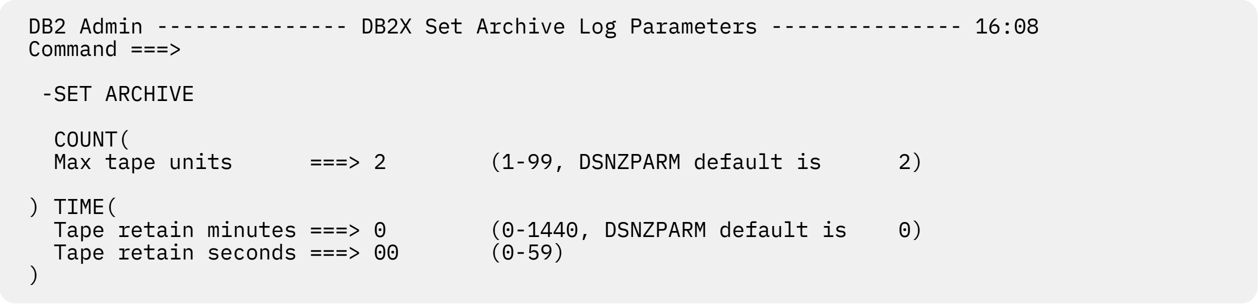
You can set the upper limit for the number of and the deallocation time of tape units for the archive log.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option LS, and press Enter.

   The **Set Archive Log Parameters (ADB2ZLSS)** panel panel is displayed, as shown in the following figure.



```
 DB2 Admin --------------- DB2X Set Archive Log Parameters --------------- 16:08
 Command ===>

  -SET ARCHIVE

   COUNT(
   Max tape units      ===> 2        (1-99, DSNZPARM default is      2)

 ) TIME(
   Tape retain minutes ===> 0        (0-1440, DSNZPARM default is    0)
   Tape retain seconds ===> 00       (0-59)
 )
```

*Figure 531. **Set Archive Log Parameters (ADB2ZLSS)** panel*

3. Specify values for the following fields:

   - **Max tape units**
   - **Tape retain minutes**
   - **Tape retain seconds**
4. Press Enter.

   Db2 Admin Tool issues the Db2 command SET ARCHIVE with the parameter settings that you specified. The information Db2 Admin Tool returns to you from the command is in ISPF browse format.

Archiving the current Db2 log

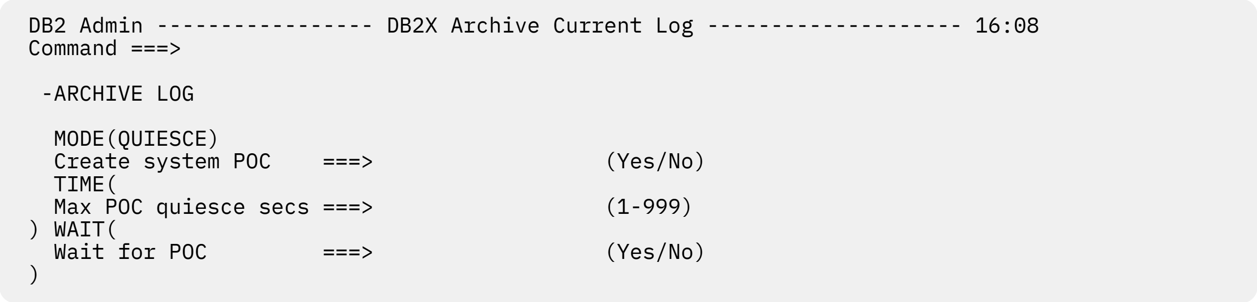
You can archive the current Db2 log.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option LA, and press Enter.

   The **Archive Current Log (ADB2ZLA)** panel is displayed, as shown in the following figure.



```
DB2 Admin ----------------- DB2X Archive Current Log -------------------- 16:08
Command ===>

 -ARCHIVE LOG

  MODE(QUIESCE)
  Create system POC    ===>                  (Yes/No)
  TIME(
  Max POC quiesce secs ===>                  (1-999)
) WAIT(
  Wait for POC         ===>                  (Yes/No)
)
```

*Figure 532. **Archive Current Log (ADB2ZLA)** panel*

3. Enter the appropriate keywords and parameters on the panel and press Enter.

   Db2 Admin Tool issues the DB2 -ARCHIVE LOG command. The command response that Db2 Admin Tool returns to is displayed in an ISPF browse session.

Displaying log information

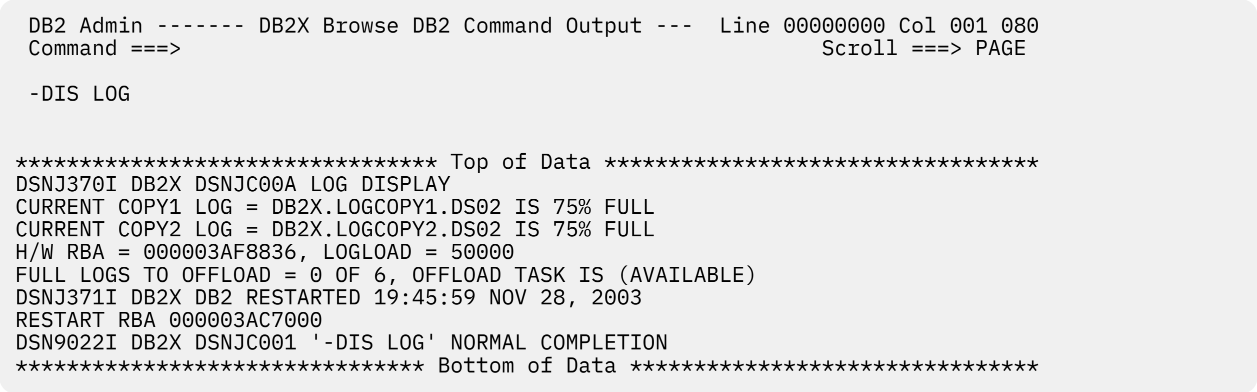
You can display Db2 log information to view the checkpoint frequency and the status of any offload tasks.

Procedure

To display log information:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option LI, and press Enter.

   The **Display Log Information (ADB2DB2O)** panel displays the result of the Db2 command DISPLAY LOG :



```
 DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                 Scroll ===> PAGE

 -DIS LOG


********************************* Top of Data **********************************
DSNJ370I DB2X DSNJC00A LOG DISPLAY
CURRENT COPY1 LOG = DB2X.LOGCOPY1.DS02 IS 75% FULL
CURRENT COPY2 LOG = DB2X.LOGCOPY2.DS02 IS 75% FULL
H/W RBA = 000003AF8836, LOGLOAD = 50000
FULL LOGS TO OFFLOAD = 0 OF 6, OFFLOAD TASK IS (AVAILABLE)
DSNJ371I DB2X DB2 RESTARTED 19:45:59 NOV 28, 2003
RESTART RBA 000003AC7000
DSN9022I DB2X DSNJC001 '-DIS LOG' NORMAL COMPLETION
******************************** Bottom of Data ********************************
```

*Figure 533. **Display Log Information (ADB2DB2O)** panel*

Related information

DISPLAY LOG (Db2 12 for z/OS documentation)

Changing Db2 system checkpoint frequency

You can change how frequently Db2 should perform a system checkpoint.

About this task

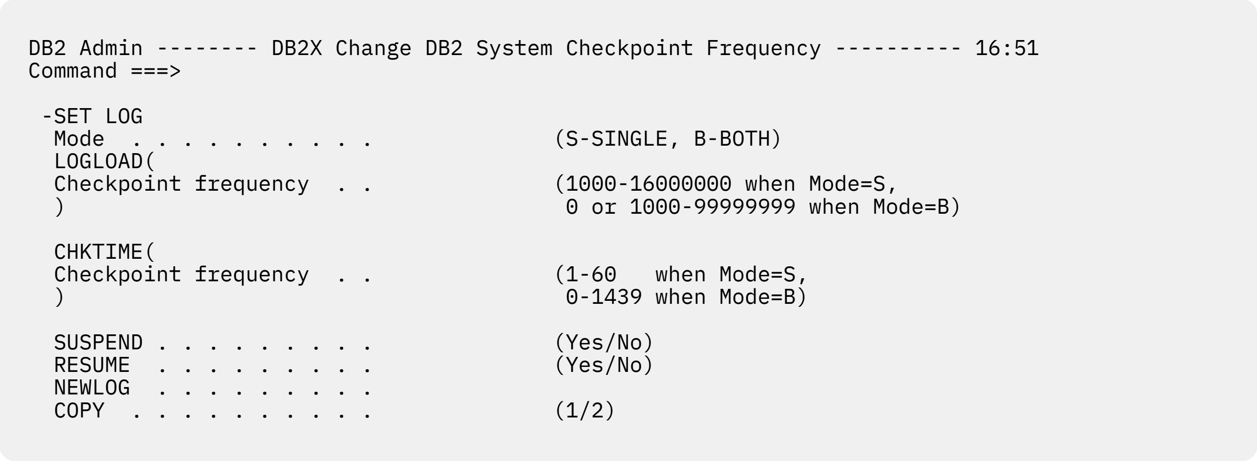
To change how frequently Db2 should perform a system checkpoint (in terms of number of Db2 log records):

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option LZ, and press Enter.

   The **Change DB2 System Checkpoint Frequency (ADB2ZLZ)** panel is displayed, as shown in the following figure.



```
DB2 Admin -------- DB2X Change DB2 System Checkpoint Frequency ---------- 16:51
Command ===>

 -SET LOG
  Mode  . . . . . . . . . .              (S-SINGLE, B-BOTH)
  LOGLOAD(
  Checkpoint frequency  . .              (1000-16000000 when Mode=S,
  )                                       0 or 1000-99999999 when Mode=B)

  CHKTIME(
  Checkpoint frequency  . .              (1-60   when Mode=S,
  )                                       0-1439 when Mode=B)

  SUSPEND . . . . . . . . .              (Yes/No)
  RESUME  . . . . . . . . .              (Yes/No)
  NEWLOG  . . . . . . . . .
  COPY  . . . . . . . . . .              (1/2)
```

*Figure 534. **Change DB2 System Checkpoint Frequency (ADB2ZLZ)** panel*

3. Enter the appropriate keywords and parameters on the panel and press Enter.

   Db2 Admin Tool issues the Db2 SET LOG command. The information Db2 Admin Tool returns to you from the command is in ISPF browse format.

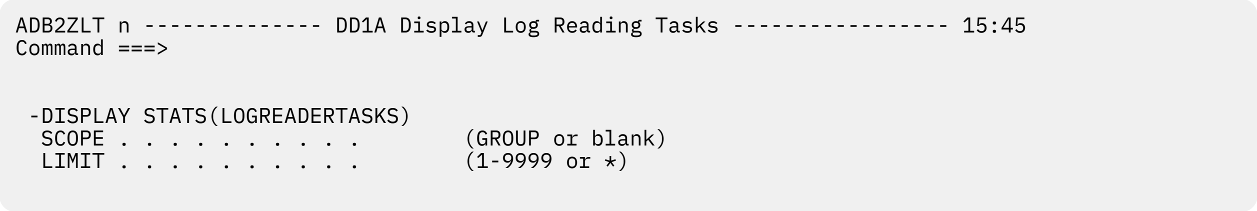
Displaying log reading tasks

You can display statistics about any log reading tasks that are currently running.

Procedure

To display log reading tasks:

1. On the **DB2 Administration Menu (ADB2)** panel, specify option Z, and press Enter.
2. On the **System Administration (ADB2Z)** panel, specify option LT, and press Enter.
3. On the **Display Log Reading Tasks (ADB2ZLT)** panel, enter any additional parameters as needed:



```
ADB2ZLT n -------------- DD1A Display Log Reading Tasks ----------------- 15:45
Command ===>


 -DISPLAY STATS(LOGREADERTASKS)
  SCOPE . . . . . . . . . .        (GROUP or blank)
  LIMIT . . . . . . . . . .        (1-9999 or *)
```

*Figure 535. **Display Log Reading Tasks (ADB2ZLT)** panel*

4. Press Enter.

   Db2 Admin Tool issues the Db2 command DIS STATS(LOGREADERTASKS).

Related information

-DISPLAY STATS (Db2) (Db2 12 for z/OS documentation)

Communications settings

Db2 uses communication settings that you can display or update.

These settings are stored in communication database (CDB) tables (SYSIBM.xxx).

Displaying or updating the LOCATIONS table

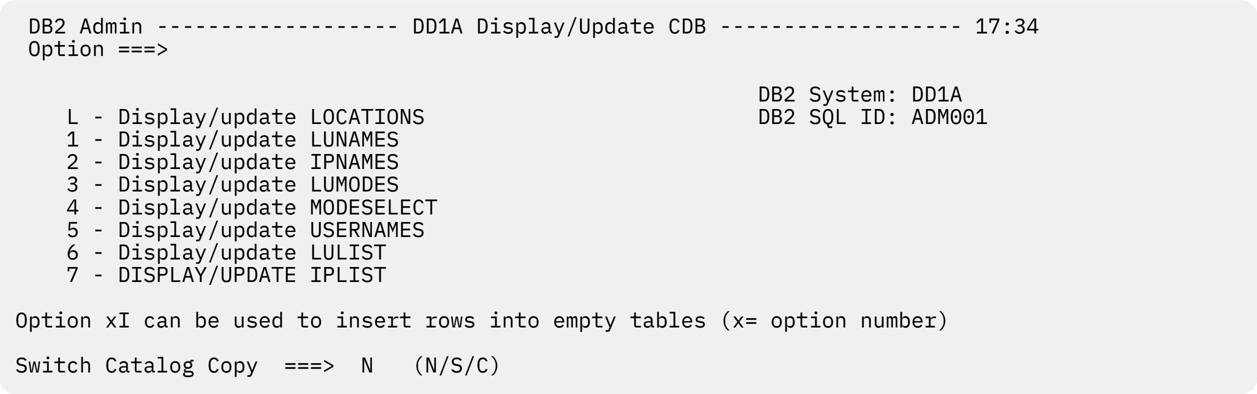
Use the **Display/Update LOCATIONS** panel to update the LOCATIONS table.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option DU, and press Enter.

   The **Display/Update CDB** panel is displayed, as shown in the following figure.

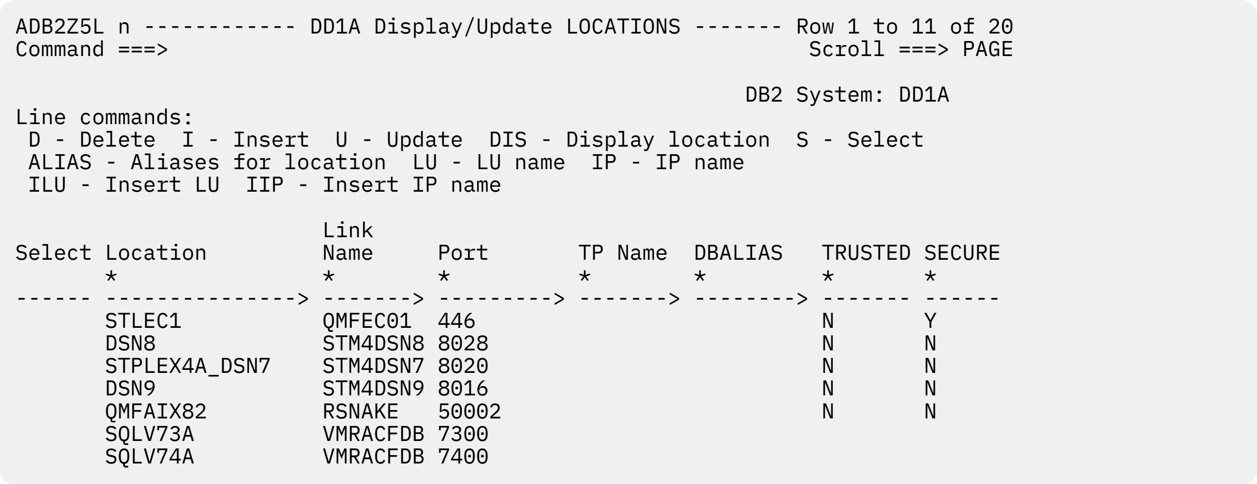


```
 DB2 Admin ------------------- DD1A Display/Update CDB ------------------- 17:34
 Option ===>
                                                        DB2 System: DD1A
    L - Display/update LOCATIONS                        DB2 SQL ID: ADM001
    1 - Display/update LUNAMES
    2 - Display/update IPNAMES
    3 - Display/update LUMODES
    4 - Display/update MODESELECT
    5 - Display/update USERNAMES
    6 - Display/update LULIST
    7 - DISPLAY/UPDATE IPLIST

Option xI can be used to insert rows into empty tables (x= option number)

Switch Catalog Copy  ===>  N   (N/S/C)
```

Figure 536. Display/Update CDB panel (ADB2Z5)

3. Select L, and press Enter.

   The **Display/Update LOCATIONS** panel is displayed, shown in the following figure.



```
ADB2Z5L n ------------ DD1A Display/Update LOCATIONS ------- Row 1 to 11 of 20
Command ===>                                                  Scroll ===> PAGE

                                                       DB2 System: DD1A
Line commands:
 D - Delete  I - Insert  U - Update  DIS - Display location  S - Select
 ALIAS - Aliases for location  LU - LU name  IP - IP name
 ILU - Insert LU  IIP - Insert IP name

                        Link
Select Location         Name     Port       TP Name  DBALIAS   TRUSTED SECURE
       *                *        *          *        *         *       *
------ ---------------> -------> ---------> -------> --------> ------- ------
       STLEC1           QMFEC01  446                           N       Y
       DSN8             STM4DSN8 8028                          N       N
       STPLEX4A_DSN7    STM4DSN7 8020                          N       N
       DSN9             STM4DSN9 8016                          N       N
       QMFAIX82         RSNAKE   50002                         N       N
       SQLV73A          VMRACFDB 7300
       SQLV74A          VMRACFDB 7400
```

Figure 537. Display/Update LOCATIONS panel (ADB2Z5L)

4. Optional: Update the table by using the following line commands, and press Enter.

D
: Deletes the row.

I
: Inserts a new row. Row values can be entered on the next panel.

U
: Updates the row. Row values can be changed on the next panel.

Displaying or updating the LUNAMES table

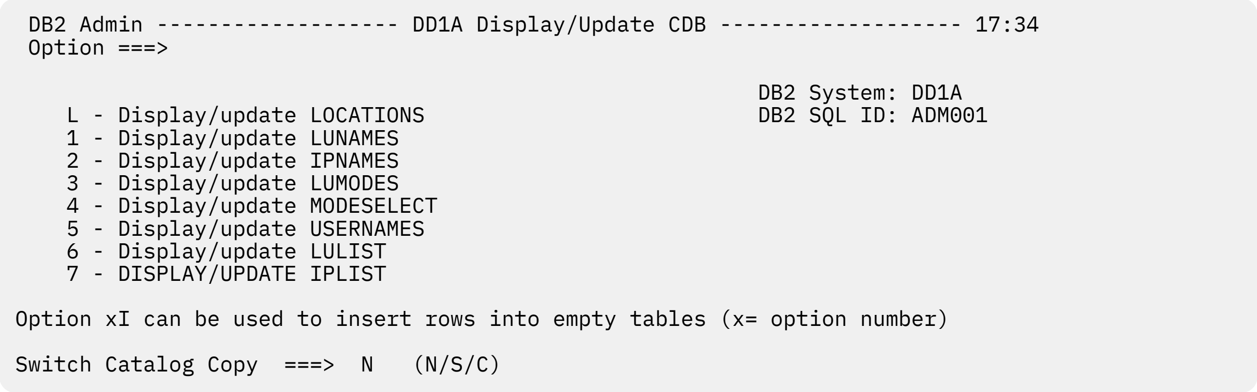
Use the **Display/Update LUNAMES** panel to update the LUNAMES table.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option DU, and press Enter.

   The **Display/Update CDB** panel is displayed, as shown in the following figure.

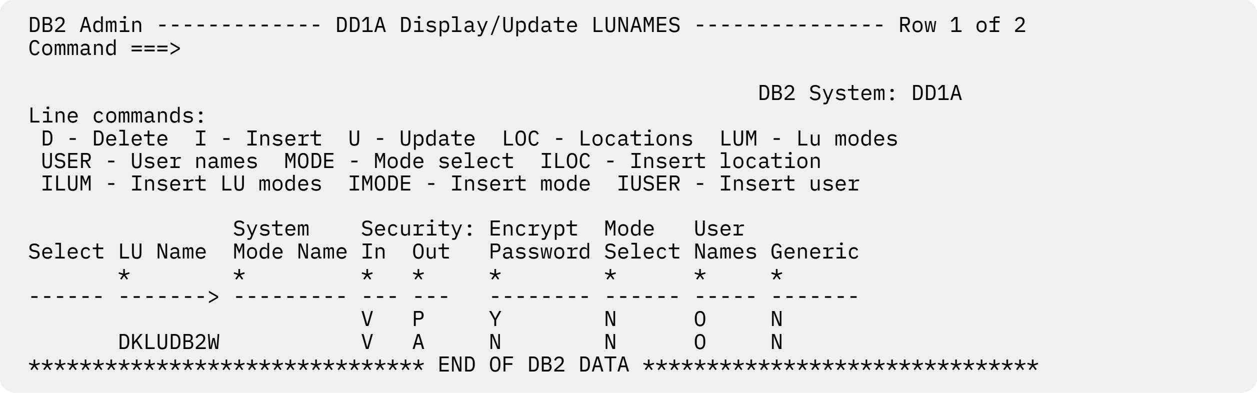


```
 DB2 Admin ------------------- DD1A Display/Update CDB ------------------- 17:34
 Option ===>

                                                          DB2 System: DD1A
    L - Display/update LOCATIONS                          DB2 SQL ID: ADM001
    1 - Display/update LUNAMES
    2 - Display/update IPNAMES
    3 - Display/update LUMODES
    4 - Display/update MODESELECT
    5 - Display/update USERNAMES
    6 - Display/update LULIST
    7 - DISPLAY/UPDATE IPLIST

Option xI can be used to insert rows into empty tables (x= option number)

Switch Catalog Copy  ===>  N   (N/S/C)
```

Figure 538. Display/Update CDB panel (ADB2Z5)

3. Select 1, and press Enter.

   The **Display/Update LUNAMES** panel is displayed, shown in the following figure.



```
DB2 Admin ------------- DD1A Display/Update LUNAMES --------------- Row 1 of 2
Command ===>

                                                         DB2 System: DD1A
Line commands:
 D - Delete  I - Insert  U - Update  LOC - Locations  LUM - Lu modes
 USER - User names  MODE - Mode select  ILOC - Insert location
 ILUM - Insert LU modes  IMODE - Insert mode  IUSER - Insert user

                System    Security: Encrypt  Mode   User
Select LU Name  Mode Name In  Out   Password Select Names Generic
       *        *         *   *     *        *      *     *
------ -------> --------- --- ---   -------- ------ ----- -------
                          V   P     Y        N      O     N
       DKLUDB2W           V   A     N        N      O     N
******************************* END OF DB2 DATA *******************************
```

Figure 539. Display/Update LUNAMES panel (ADB2Z51)

4. Optional: Update the table by using the following line commands, and press Enter.

   D
   : Deletes the row.

   I
   : Inserts a new row. Row values can be entered on the next panel.

U
: Updates the row. Row values can be changed on the next panel.

Displaying or updating the IPNAMES table

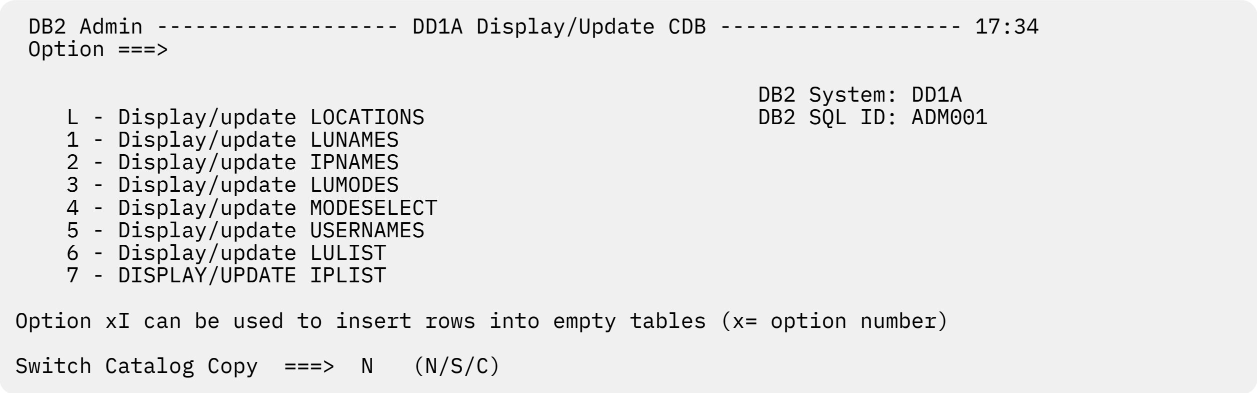
Use the **Display/Update IPNAMES** panel to update the IPNAMES table.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel , specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option DU, and press Enter.

   The **Display/Update CDB** panel is displayed, as shown in the following figure.

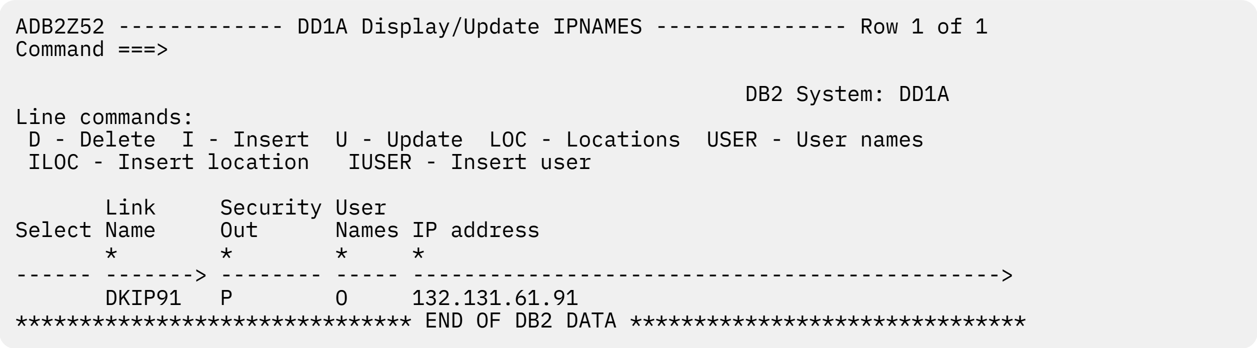


```
 DB2 Admin ------------------- DD1A Display/Update CDB ------------------- 17:34
 Option ===>

                                                          DB2 System: DD1A
    L - Display/update LOCATIONS                          DB2 SQL ID: ADM001
    1 - Display/update LUNAMES
    2 - Display/update IPNAMES
    3 - Display/update LUMODES
    4 - Display/update MODESELECT
    5 - Display/update USERNAMES
    6 - Display/update LULIST
    7 - DISPLAY/UPDATE IPLIST

Option xI can be used to insert rows into empty tables (x= option number)

Switch Catalog Copy  ===>  N   (N/S/C)
```

*Figure 540. **Display/Update CDB** panel (ADB2Z5)*

3. Select 2, and press Enter.

   The **Display/Update LOCATIONS** panel is displayed, shown in the following figure.



```
ADB2Z52 ------------- DD1A Display/Update IPNAMES --------------- Row 1 of 1
Command ===>

                                                         DB2 System: DD1A
Line commands:
 D - Delete  I - Insert  U - Update  LOC - Locations  USER - User names
 ILOC - Insert location   IUSER - Insert user

       Link     Security User
Select Name     Out      Names IP address
       *        *        *     *
------ -------> -------- ----- ---------------------------------------------->
       DKIP91   P        O     132.131.61.91
******************************* END OF DB2 DATA *******************************
```

Figure 541. Display/Update IPNAMES panel (ADB2Z52)

4. Optional: Update the table by using the following line commands, and press Enter.

   D
   : Deletes the row.

   I
   : Inserts a new row. Row values can be entered on the next panel.

   U
   : Updates the row. Row values can be changed on the next panel.

Displaying or updating the LUMODES table

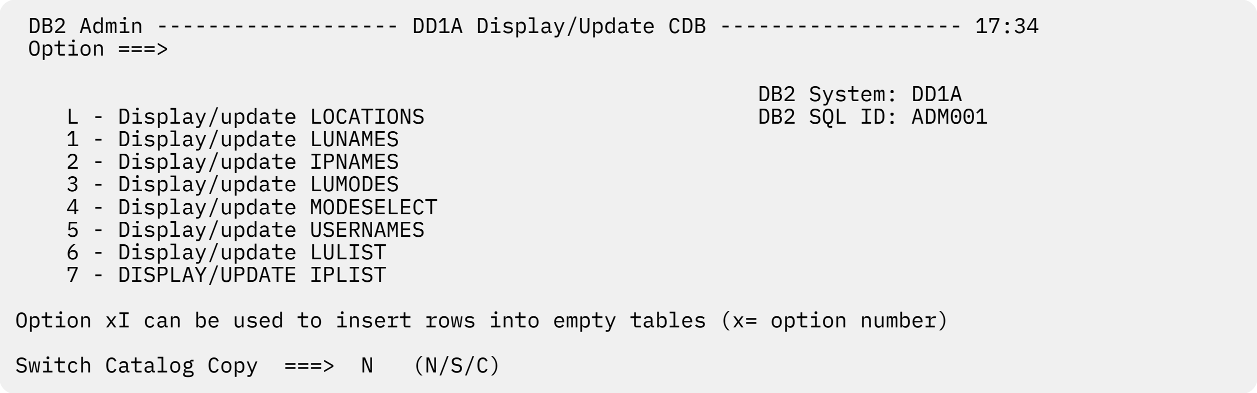
Use the **Display/Update LUMODES modes** panel to update the update the LUMODES table.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option DU, and press Enter.

   The **Display/Update CDB** panel is displayed, as shown in the following figure.

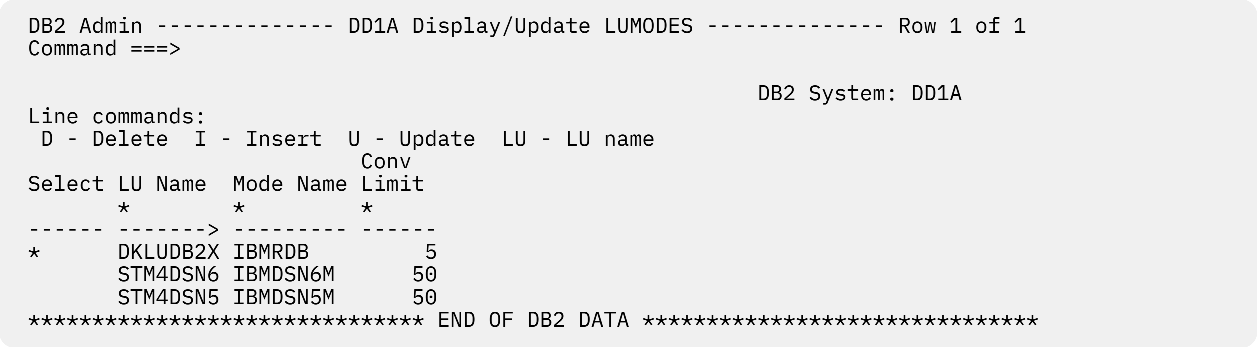


```
 DB2 Admin ------------------- DD1A Display/Update CDB ------------------- 17:34
 Option ===>

                                                          DB2 System: DD1A
    L - Display/update LOCATIONS                          DB2 SQL ID: ADM001
    1 - Display/update LUNAMES
    2 - Display/update IPNAMES
    3 - Display/update LUMODES
    4 - Display/update MODESELECT
    5 - Display/update USERNAMES
    6 - Display/update LULIST
    7 - DISPLAY/UPDATE IPLIST

Option xI can be used to insert rows into empty tables (x= option number)

Switch Catalog Copy  ===>  N   (N/S/C)
```

Figure 542. Display/Update CDB panel (ADB2Z5)

3. Select 3, and press Enter.

   The **Display/Update LUMODES modes** panel is displayed, shown in the following figure.



```
 DB2 Admin -------------- DD1A Display/Update LUMODES -------------- Row 1 of 1
 Command ===>

                                                          DB2 System: DD1A
 Line commands:
  D - Delete  I - Insert  U - Update  LU - LU name
                          Conv
 Select LU Name  Mode Name Limit
        *        *         *
 ------ -------> --------- ------
 *      DKLUDB2X IBMRDB         5
        STM4DSN6 IBMDSN6M      50
        STM4DSN5 IBMDSN5M      50
 ******************************* END OF DB2 DATA *******************************
```

Figure 543. Display/Update LUMODES panel (ADB2Z53)

4. Optional: Update the table by using the following line commands, and press Enter.

   D

   Deletes the row.

   I

   Inserts a new row. Row values can be entered on the next panel.

   U

   Updates the row. Row values can be changed on the next panel.

Displaying or updating the MODESELECT table

Use the **Display/Update MODESELECT** panel to update the MODESELECT table.

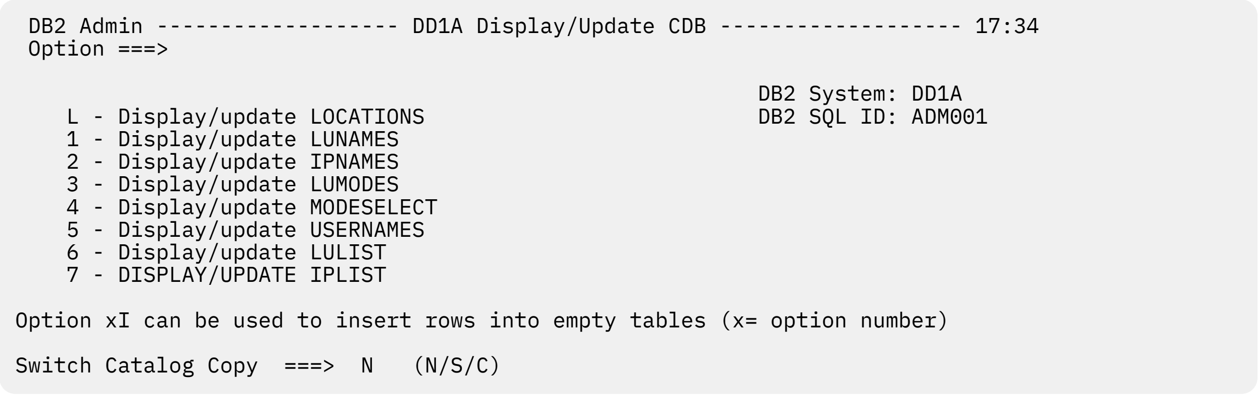
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

The **System Administration (ADB2Z)** panel is displayed.

2. Specify option DU, and press Enter.

   The **Display/Update CDB** panel is displayed, as shown in the following figure.

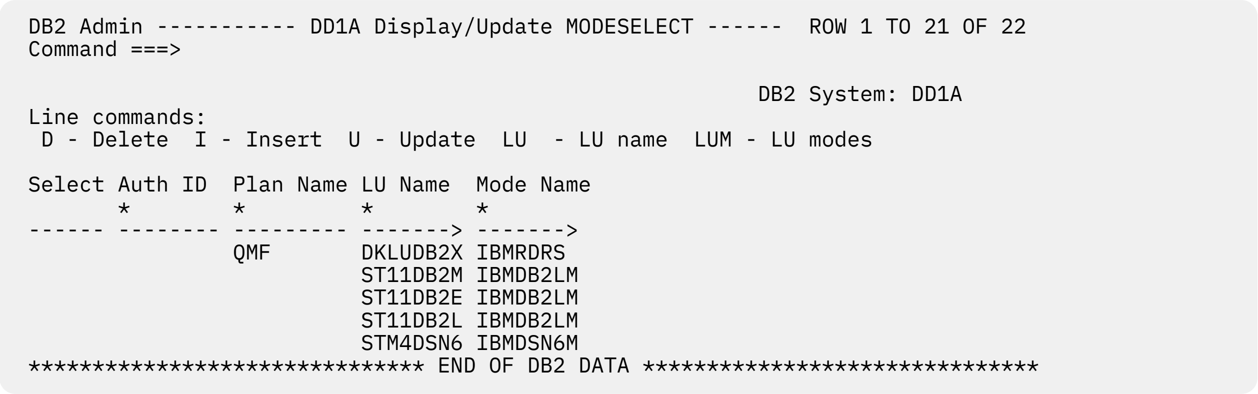


```
 DB2 Admin ------------------- DD1A Display/Update CDB ------------------- 17:34
 Option ===>
                                                         DB2 System: DD1A
    L - Display/update LOCATIONS                         DB2 SQL ID: ADM001
    1 - Display/update LUNAMES
    2 - Display/update IPNAMES
    3 - Display/update LUMODES
    4 - Display/update MODESELECT
    5 - Display/update USERNAMES
    6 - Display/update LULIST
    7 - DISPLAY/UPDATE IPLIST

Option xI can be used to insert rows into empty tables (x= option number)

Switch Catalog Copy  ===>  N   (N/S/C)
```

Figure 544. Display/Update CDB panel (ADB2Z5)

3. Select 4, and press Enter.

   The **Display/Update MODESELECT** panel is displayed, shown in the following figure.



```
 DB2 Admin ----------- DD1A Display/Update MODESELECT ------  ROW 1 TO 21 OF 22
 Command ===>
                                                         DB2 System: DD1A
 Line commands:
  D - Delete  I - Insert  U - Update  LU  - LU name  LUM - LU modes

 Select Auth ID  Plan Name LU Name  Mode Name
        *        *         *        *
 ------ -------- --------- -------> ------->
                 QMF       DKLUDB2X IBMRDRS
                           ST11DB2M IBMDB2LM
                           ST11DB2E IBMDB2LM
                           ST11DB2L IBMDB2LM
                           STM4DSN6 IBMDSN6M
 ****************************** END OF DB2 DATA *******************************
```

Figure 545. Display/Update MODESELECT panel (ADB2Z54)

4. Optional: Update the table by using the following line commands, and press Enter.

   D

   Deletes the row.

   I

   Inserts a new row. Row values can be entered on the next panel.

   U

   Updates the row. Row values can be changed on the next panel.

Displaying or updating the USERNAMES table

Use the **Display/Update USERNAMES** panel to update the USERNAMES table.

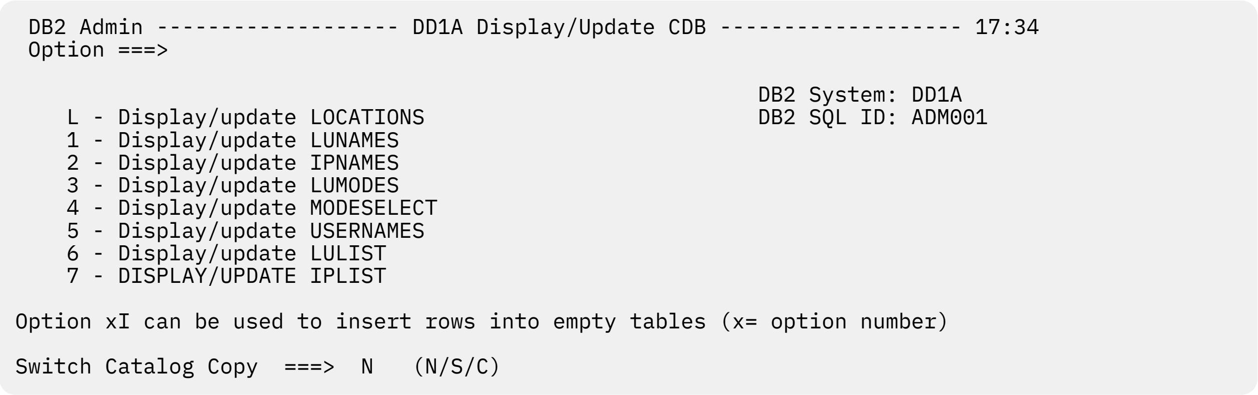
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.

2. Specify option DU, and press Enter.

   The **Display/Update CDB** panel is displayed, as shown in the following figure.

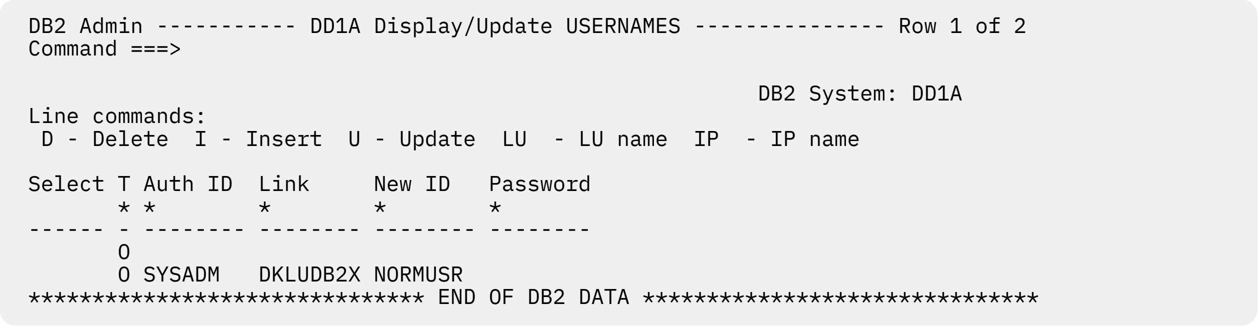


```
 DB2 Admin ------------------- DD1A Display/Update CDB ------------------- 17:34
 Option ===>

                                                          DB2 System: DD1A
    L - Display/update LOCATIONS                          DB2 SQL ID: ADM001
    1 - Display/update LUNAMES
    2 - Display/update IPNAMES
    3 - Display/update LUMODES
    4 - Display/update MODESELECT
    5 - Display/update USERNAMES
    6 - Display/update LULIST
    7 - DISPLAY/UPDATE IPLIST

Option xI can be used to insert rows into empty tables (x= option number)

Switch Catalog Copy  ===>  N   (N/S/C)
```

Figure 546. Display/Update CDB panel (ADB2Z5)

3. Select 5, and press Enter.

   The **Display/Update USERNAMES** panel is displayed, shown in the following figure.



```
 DB2 Admin ----------- DD1A Display/Update USERNAMES --------------- Row 1 of 2
 Command ===>

                                                          DB2 System: DD1A
 Line commands:
  D - Delete  I - Insert  U - Update  LU  - LU name  IP  - IP name

 Select T Auth ID  Link     New ID   Password
        * *        *        *        *
 ------ - -------- -------- -------- --------
        O
        O SYSADM   DKLUDB2X NORMUSR
 ******************************* END OF DB2 DATA *******************************
```

Figure 547. Display/Update USERNAMES panel (ADB2Z55)

4. Optional: Update the table by using the following line commands, and press Enter.

   D

   Deletes the row.

   I

   Inserts a new row. Row values can be entered on the next panel.

   U

   Updates the row. Row values can be changed on the next panel.

Displaying or updating the LULIST table

Use the **Display/Update LULIST** panel to update the LULIST table.

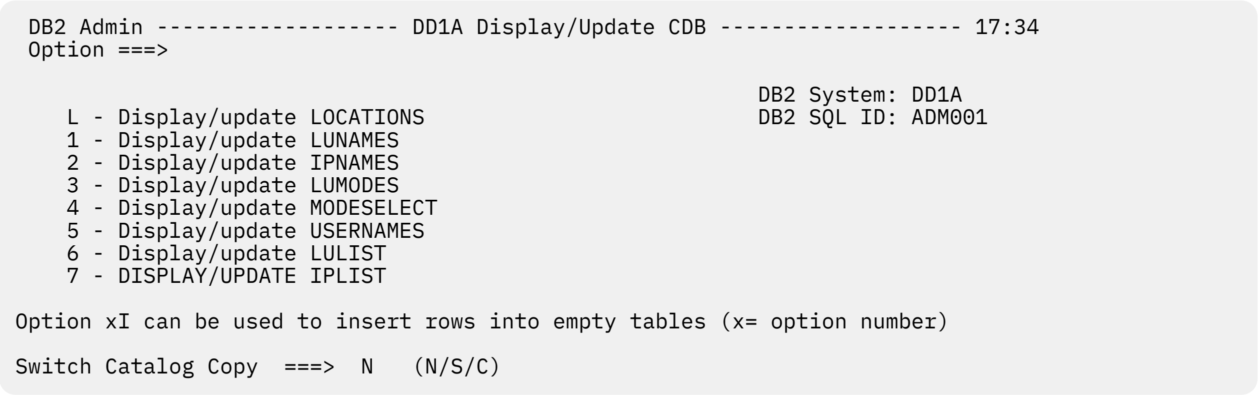
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.

2. Specify option DU, and press Enter.

   The **Display/Update CDB** panel is displayed, as shown in the following figure.

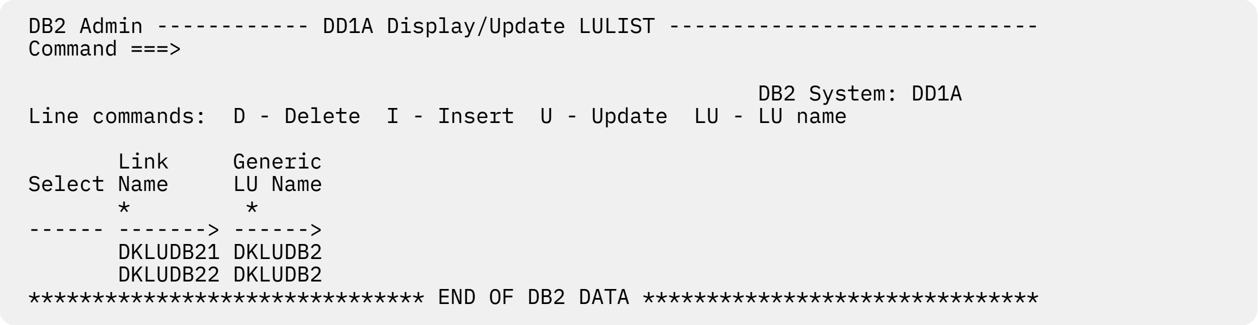


```
 DB2 Admin ------------------- DD1A Display/Update CDB ------------------- 17:34
 Option ===>
                                                         DB2 System: DD1A
    L - Display/update LOCATIONS                         DB2 SQL ID: ADM001
    1 - Display/update LUNAMES
    2 - Display/update IPNAMES
    3 - Display/update LUMODES
    4 - Display/update MODESELECT
    5 - Display/update USERNAMES
    6 - Display/update LULIST
    7 - DISPLAY/UPDATE IPLIST

Option xI can be used to insert rows into empty tables (x= option number)

Switch Catalog Copy  ===>  N   (N/S/C)
```

Figure 548. Display/Update CDB panel (ADB2Z5)

3. Select 6, and press Enter.

   The **Display/Update LULIST** panel is displayed, shown in the following figure.



```
 DB2 Admin ------------ DD1A Display/Update LULIST ----------------------------
 Command ===>
                                                         DB2 System: DD1A
 Line commands:  D - Delete  I - Insert  U - Update  LU - LU name

        Link     Generic
 Select Name     LU Name
        *         *
 ------ -------> ------>
        DKLUDB21 DKLUDB2
        DKLUDB22 DKLUDB2
 ******************************* END OF DB2 DATA *******************************
```

Figure 549. Display/Update LULIST panel (ADB2Z56)

4. Optional: Update the table by using the following line commands, and press Enter.

   D

   Deletes the row.

   I

   Inserts a new row. Row values can be entered on the next panel.

   U

   Updates the row. Row values can be changed on the next panel.

Displaying DDF

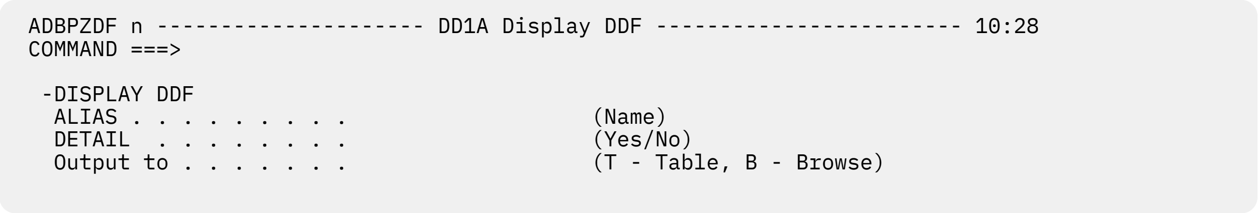
You can display the status and configuration of the distributed data facility (DDF) for your Db2 subsystem.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option DF, and press Enter.

   The **Display DDF (ADBPZDF)** panel is displayed, as shown in the following figure.



```
 ADBPZDF n --------------------- DD1A Display DDF ------------------------ 10:28
 COMMAND ===>

  -DISPLAY DDF
   ALIAS . . . . . . . . .                   (Name)
   DETAIL  . . . . . . . .                   (Yes/No)
   Output to . . . . . . .                   (T - Table, B - Browse)
```

*Figure 550. **Display DDF (ADBPZDF)** panel*

Panel ADBPZDF helps you to construct a Db2 DISPLAY DDF command, which displays the DDF information in a report. You can specify the following options for the -DISPLAY DDF command:

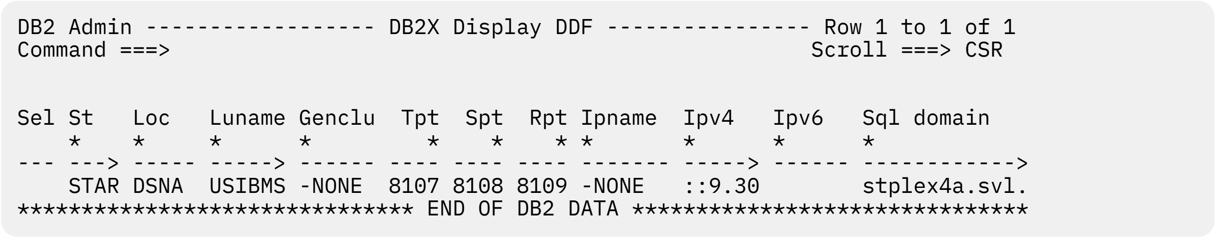
ALIAS
Displays information specific to the DDF location alias specified by **`alias-name`**.

DETAIL
Specifies whether to display additional statistics and configuration information.

Output to
Specifies where to store the result of the DISPLAY DDF command. Select T (Table) to display the results in an ISPF table, or B (Browse) to display a report.

3. Specify values for the **Alias** and **Detail** fields.
4. Choose one of the following options, and complete the steps:

Write output to a table
In the **Output to** field, specify T, and press Enter. The **Display DDF** panel is displayed, as shown in the following figure.



```
DB2 Admin ------------------ DB2X Display DDF ---------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> CSR


Sel St   Loc   Luname Genclu  Tpt  Spt  Rpt Ipname  Ipv4   Ipv6   Sql domain
    *    *     *      *         *    *    * *       *      *      *
--- ---> ----- -----> ------ ---- ---- ---- ------- -----> ------ ------------>
    STAR DSNA  USIBMS -NONE  8107 8108 8109 -NONE   ::9.30        stplex4a.svl.
******************************* END OF DB2 DATA *******************************
```

*Figure 551. **Display DDF** panel*

The following columns are on this panel:

Sel
You can use the / line command to view details for each of the fields on the panel.

St
Displays the DDF status.

Loc
Displays the location name of the DDF as it is recorded in the bootstrap data set (BSDS).

Luname
Displays the DDF LU name as recorded in the BSDS.

Genclu
Displays the DDF generic LU name as recorded in the BSDS.

Tpt
Displays the TCP/IP port number for the SQL listener as recorded in the BSDS.

Spt
Displays the TCP/IP port number for the secure SQL listener as recorded in the BSDS.

Rpt
Displays the TCP/IP port number for the two-phase commit resynchronization (resync) listener, as recorded in the BSDS.

Ipname
Displays the IPNAME value as recorded in the BSDS.

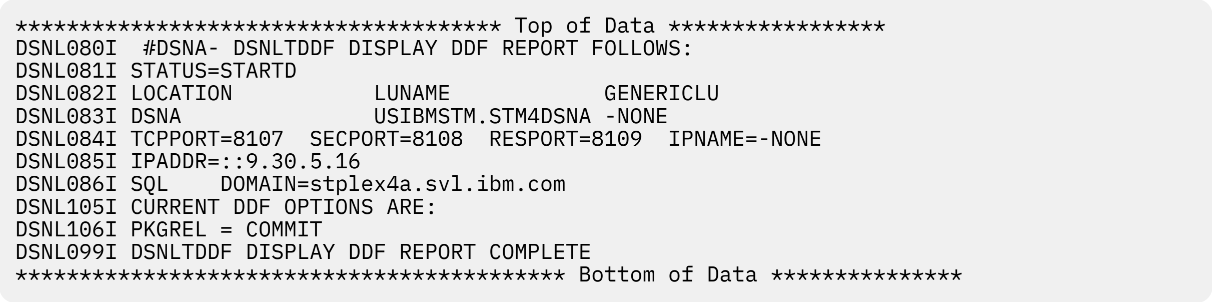
Ipv4
Displays the IP address of the DDF using IPV4 format.

Ipv6
Displays the IP address of the DDF using IPV6 format.

Sqldomain
Displays the TCP/IP domain name that is associated with the DDF.

Browse the DDF information

In the **Output to** field, specify B, and press Enter. The **Display DDF (ADB2DB2O)** panel is displayed, as shown in the following figure.The report is displayed, as shown in the following example figure.



```
************************************** Top of Data *****************
DSNL080I  #DSNA- DSNLTDDF DISPLAY DDF REPORT FOLLOWS:
DSNL081I STATUS=STARTD
DSNL082I LOCATION           LUNAME            GENERICLU
DSNL083I DSNA               USIBMSTM.STM4DSNA -NONE
DSNL084I TCPPORT=8107  SECPORT=8108  RESPORT=8109  IPNAME=-NONE
DSNL085I IPADDR=::9.30.5.16
DSNL086I SQL    DOMAIN=stplex4a.svl.ibm.com
DSNL105I CURRENT DDF OPTIONS ARE:
DSNL106I PKGREL = COMMIT
DSNL099I DSNLTDDF DISPLAY DDF REPORT COMPLETE
******************************************* Bottom of Data ***************
```

Figure 552. ***Display DDF (ADB2DB2O)*** *panel*

Displaying or canceling distributed threads

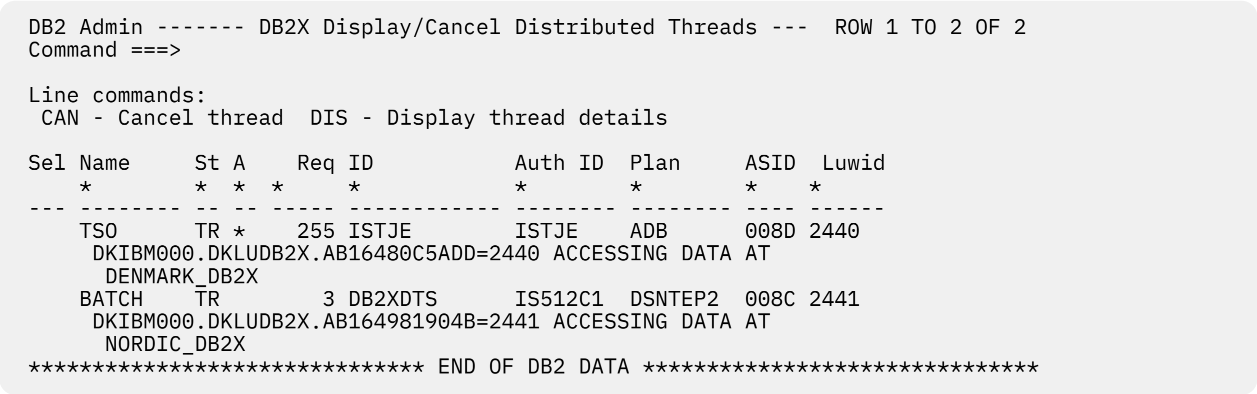
You can cancel processing for distributed data facility (DDF) threads that originate locally and access remote data, or that originate remotely and access local data.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel, is displayed.
2. Specify option DC, and press Enter.

   The **Display/Cancel Distributed Threads (ADB2ZDC2)** panel panel is displayed, as shown in the following figure.



```
DB2 Admin ------- DB2X Display/Cancel Distributed Threads ---  ROW 1 TO 2 OF 2
Command ===>

Line commands:
 CAN - Cancel thread  DIS - Display thread details

Sel Name     St A    Req ID           Auth ID  Plan     ASID  Luwid
    *        *  *  *     *            *        *        *    *
--- -------- -- -- ----- ------------ -------- -------- ---- ------
    TSO      TR *    255 ISTJE        ISTJE    ADB      008D 2440
     DKIBM000.DKLUDB2X.AB16480C5ADD=2440 ACCESSING DATA AT
      DENMARK_DB2X
    BATCH    TR        3 DB2XDTS      IS512C1  DSNTEP2  008C 2441
     DKIBM000.DKLUDB2X.AB164981904B=2441 ACCESSING DATA AT
      NORDIC_DB2X
******************************* END OF DB2 DATA *******************************
```

Figure 553. ***Display/Cancel Distributed Threads (ADB2ZDC2)*** *panel*

The following columns are available on this panel:

Sel
Input field where you enter one of the line commands listed on the panel.

Name
Connection name.

St
Connection status.

A
Active indicator.

Req
Number of Db2 requests.

ID
Correlation ID.

Auth ID
Authorization ID.

Plan
Plan name.

ASID
Address space ID.

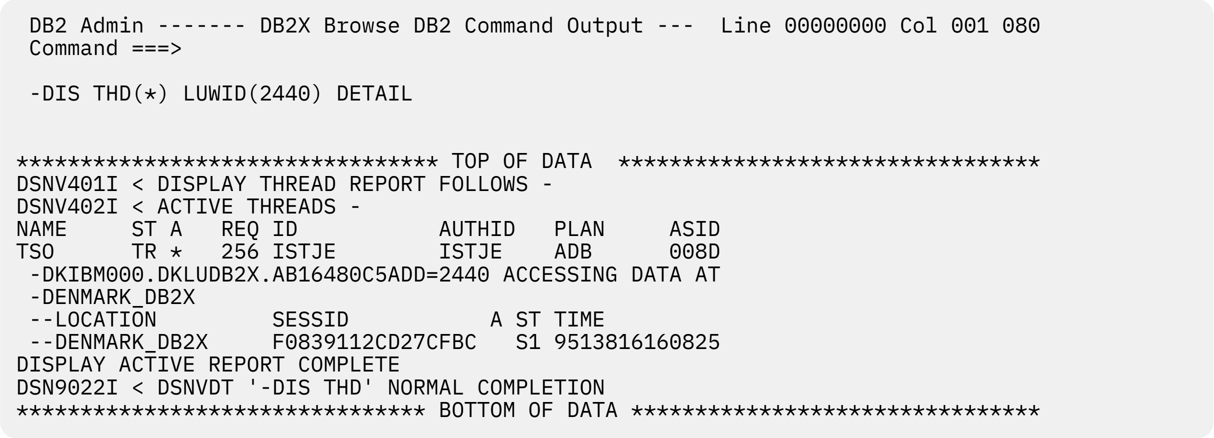
Luwid
Logical unit-of-work ID.

3. Issue one of the following line commands:

   CAN
   Cancels a thread. When you press Enter, Db2 Admin Tool issues the CANCEL DDF THREAD command.

   DIS
   Displays detailed information about a thread. When you press Enter, Db2 Admin Tool issues the Db2 DISPLAY THREAD DETAILS command.

   The following figure shows the type of information Db2 Admin Tool returns when you issue the DIS line command to display information about a thread.



```
 DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>

 -DIS THD(*) LUWID(2440) DETAIL


********************************* TOP OF DATA  *********************************
DSNV401I < DISPLAY THREAD REPORT FOLLOWS -
DSNV402I < ACTIVE THREADS -
NAME     ST A   REQ ID           AUTHID   PLAN     ASID
TSO      TR *   256 ISTJE        ISTJE    ADB      008D
 -DKIBM000.DKLUDB2X.AB16480C5ADD=2440 ACCESSING DATA AT
 -DENMARK_DB2X
 --LOCATION         SESSID           A ST TIME
 --DENMARK_DB2X     F0839112CD27CFBC   S1 9513816160825
DISPLAY ACTIVE REPORT COMPLETE
DSN9022I < DSNVDT '-DIS THD' NORMAL COMPLETION
******************************** BOTTOM OF DATA ********************************
```

*Figure 554. **Display Distributed Threads (ADB2DB2O)** panel*

The information Db2 Admin Tool returns to you from the commands is in ISPF browse format.

Displaying location details and threads

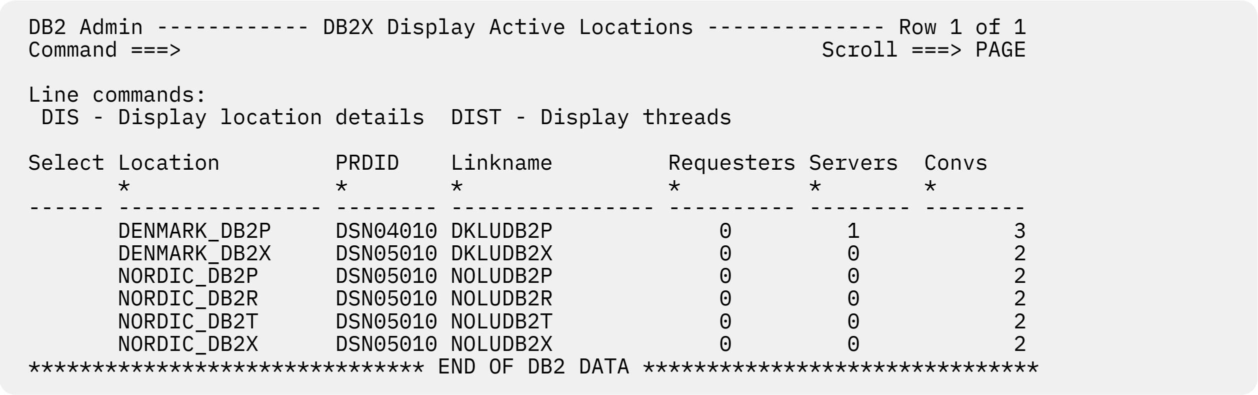
You can display statistics about threads with a distributed relationship, or display conversation information about Db2 system threads that interact with VTAM.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option DL, and press Enter.

   The **Display Active Locations (ADB2ZDL2)** panel is displayed, as shown in the following figure.



```
DB2 Admin ------------ DB2X Display Active Locations ------------- Row 1 of 1
Command ===>                                                  Scroll ===> PAGE

Line commands:
 DIS - Display location details  DIST - Display threads

Select Location         PRDID    Linkname         Requesters Servers  Convs
       *                *        *                *          *        *
------ ---------------- -------- ---------------- ---------- -------- --------
       DENMARK_DB2P     DSN04010 DKLUDB2P             0         1            3
       DENMARK_DB2X     DSN05010 DKLUDB2X             0         0            2
       NORDIC_DB2P      DSN05010 NOLUDB2P             0         0            2
       NORDIC_DB2R      DSN05010 NOLUDB2R             0         0            2
       NORDIC_DB2T      DSN05010 NOLUDB2T             0         0            2
       NORDIC_DB2X      DSN05010 NOLUDB2X             0         0            2
******************************* END OF DB2 DATA *******************************
```

Figure 555. ***Display Active Locations (ADB2ZDL2)*** *panel*

The following columns are available on this panel:

Select
Input field where you enter one of the line commands listed on the panel.

Location
Location name.

PRDID
Database product.

Linkname
LU name.

Requesters
Number of requestors.

Servers
Number of servers.

Convs
Number of conversations.

3. Issue one of the following line commands:

DIS
Displays detailed information about a thread. When you press Enter, Db2 Admin Tool issues the Db2 DISPLAY THREAD DETAILS command.

DIST
Displays the threads. When you press Enter, Db2 Admin Tool issues the Db2 DISPLAY THREAD command.

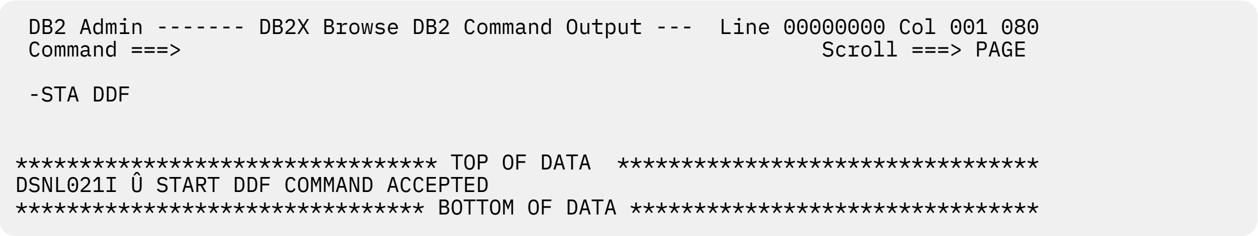
The information Db2 Admin Tool returns to you from the commands is in ISPF browse format.

Starting DDF

This procedure explains how to start the distributed data facility (DDF).

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.
   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option DT, and press Enter.
   Db2 Admin Tool issues the Db2 STA DDF command and displays the status of the command in an ISPF browse session, as shown in the following figure.



```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                                  Scroll ===> PAGE

-STA DDF


********************************* TOP OF DATA  *********************************
DSNL021I Û START DDF COMMAND ACCEPTED
******************************** BOTTOM OF DATA ********************************
```

Figure 556. Start DDF panel (ADB2DB2O)

Stopping DDF

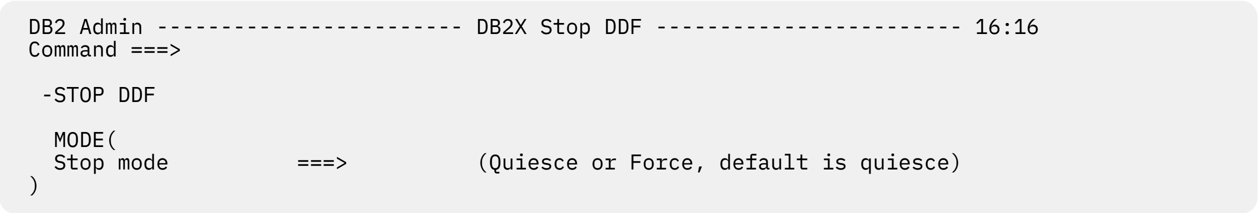
This procedure explains how to stop the distributed data facility (DDF).

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option DS, and press Enter.

   The **Stop DDF (ADB2ZDS)** panel is displayed, as shown in the following figure.



```
DB2 Admin ------------------------ DB2X Stop DDF ------------------------ 16:16
Command ===>

 -STOP DDF

  MODE(
  Stop mode          ===>          (Quiesce or Force, default is quiesce)
)
```

*Figure 557. **Stop DDF (ADB2ZDS)** panel*

3. In the **Stop Mode** field, specify `Quiesce` or `Force`, and press Enter.

   Db2 Admin Tool issues the Db2 STOP DDF command. The information Db2 Admin Tool returns to you from the command is in ISPF browse format.

Stored procedures

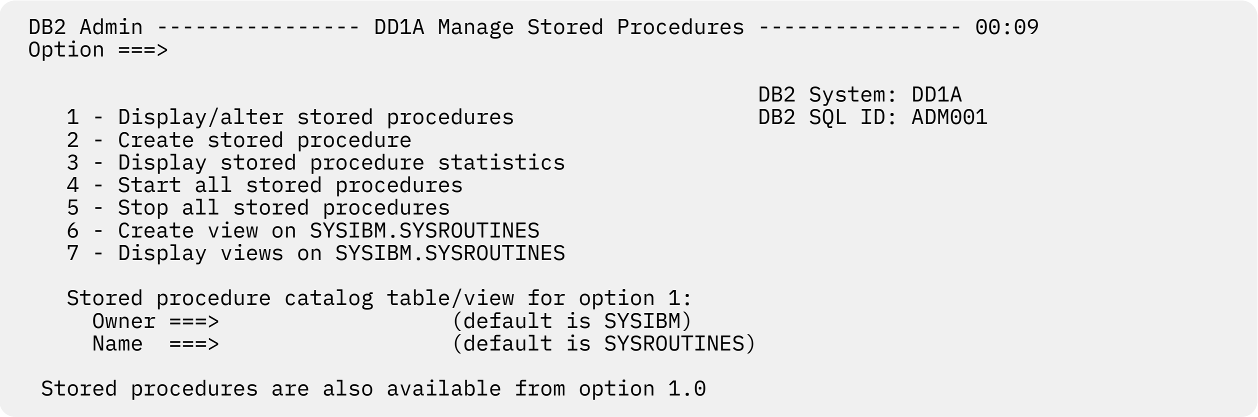
A *stored procedure* is executable code that can be called by other programs. You can create your own stored procedures to perform the functions you need. Additionally, Db2 and Db2 Admin Tool each provide some stored procedures to perform common functions.

In Db2 Admin Tool, you can create, edit, call, start and stop stored procedures from the **Manage Stored Procedures (ADB2ZP)** panel. To navigate to this panel:

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option PM, and press Enter.

   The **Manage Stored Procedures (ADB2ZP)** panel is displayed and lists the stored procedure operations that are supported by Db2 Admin Tool. The format of this panel varies depending on the version of Db2 that you are using.



```
DB2 Admin ---------------- DD1A Manage Stored Procedures ---------------- 00:09
Option ===>
                                                         DB2 System: DD1A
   1 - Display/alter stored procedures                   DB2 SQL ID: ADM001
   2 - Create stored procedure
   3 - Display stored procedure statistics
   4 - Start all stored procedures
   5 - Stop all stored procedures
   6 - Create view on SYSIBM.SYSROUTINES
   7 - Display views on SYSIBM.SYSROUTINES

   Stored procedure catalog table/view for option 1:
     Owner ===>                  (default is SYSIBM)
     Name  ===>                  (default is SYSROUTINES)

 Stored procedures are also available from option 1.0
```

*Figure 558. **Manage Stored Procedures (ADB2ZP)** panel*

Related tasks

“Defining the provided stored procedures ” on page 117
If you plan to use any of the stored procedures that are provided by Db2 Admin Tool, define those procedures during the customization process.

Related information

Procedures that are supplied with Db2 (Db2 12 for z/OS)

Creating stored procedures

In Db2 Admin Tool, you can create any of the stored procedure types that are supported by Db2 for z/OS.

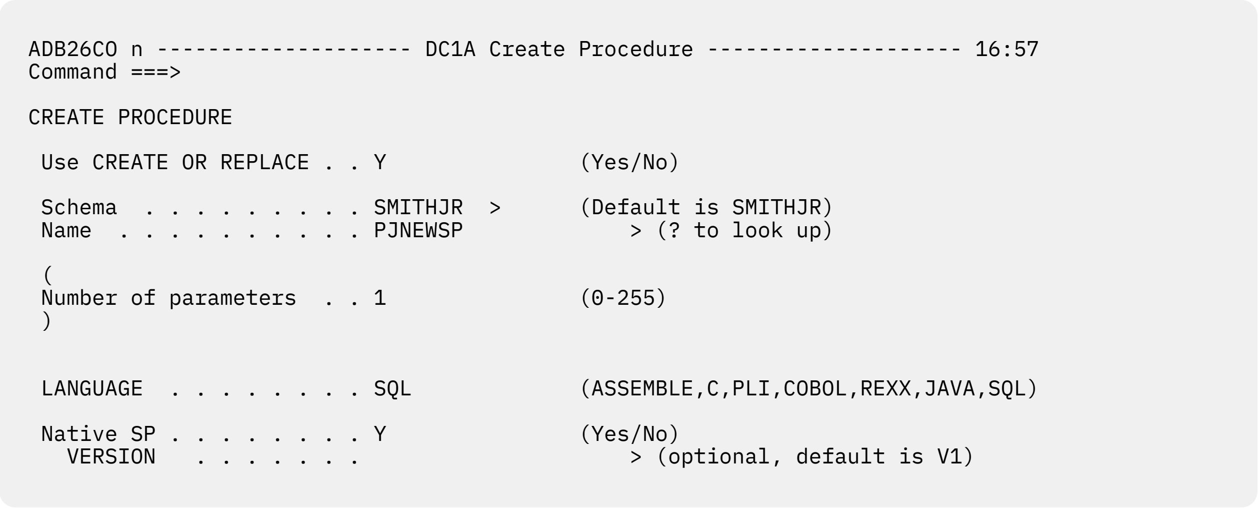
Before you begin

Ensure that the **Manage Stored Procedures (ADB2ZP)** panel is open. For instructions on how to navigate to this panel, see “Stored procedures” on page 1013.

Procedure

To create a stored procedure:

1. On the **Manage Stored Procedures (ADB2ZP)** panel, specify option 2, and press Enter.
2. On the **Create Procedure (ADB26CO)** panel, specify values for the required fields, and press Enter:



```
ADB26CO n -------------------- DC1A Create Procedure -------------------- 16:57
Command ===>

CREATE PROCEDURE

 Use CREATE OR REPLACE . . Y               (Yes/No)

 Schema  . . . . . . . . . SMITHJR  >      (Default is SMITHJR)
 Name  . . . . . . . . . . PJNEWSP             > (? to look up)

 (
 Number of parameters  . . 1               (0-255)
 )


 LANGUAGE  . . . . . . . . SQL             (ASSEMBLE,C,PLI,COBOL,REXX,JAVA,SQL)

 Native SP . . . . . . . . Y               (Yes/No)
   VERSION   . . . . . . .                     > (optional, default is V1)
```

*Figure 559. **Create Procedure (ADB26CO)** panel*

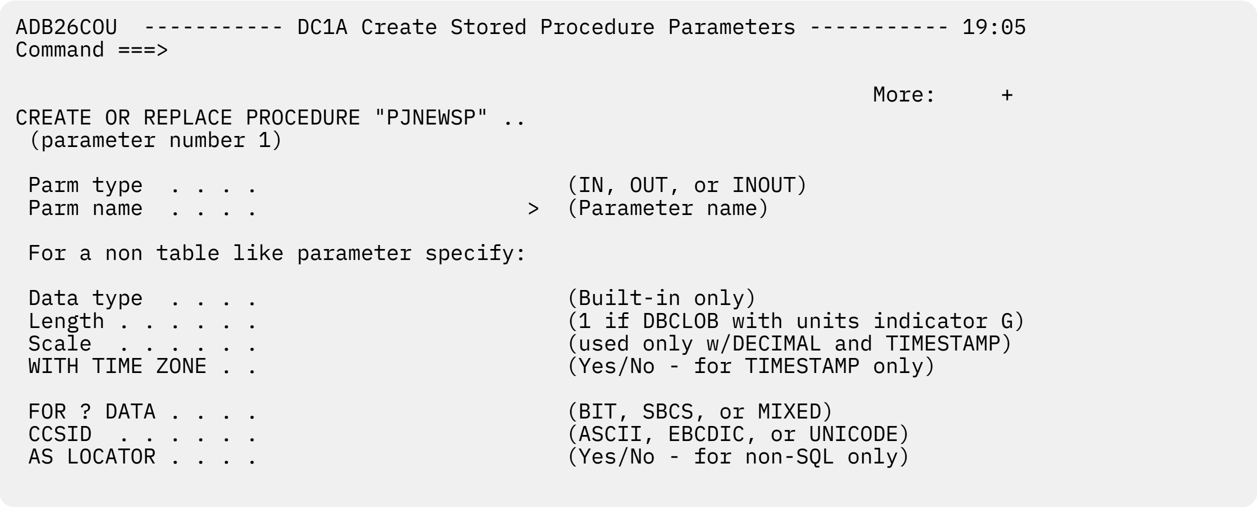
Note: The **Use CREATE OR REPLACE** field is displayed only if you are running Db2 12 and the current APPLCOMPAT value is 507 or later.

Tips:

- To create a native stored procedure, specify SQL in the **LANGUAGE** field and Y in the **Native SP** field.
- To create an external stored procedure, specify the language of the procedure body in the **LANGUAGE** field.
- To create an external SQL procedure (which is deprecated in Db2 for z/OS), specify SQL in the **LANGUAGE** field and N in the **Native SP** field.
- To cancel the process of creating a stored procedure, press End.

3. If the **Create Stored Procedure Parameters (ADB26COU)** panel is displayed, specify values for the required fields to define the first parameter, and press Enter. Repeat this step until all parameters are defined.

   This panel is displayed if you specified a number greater than zero (0) for the number of stored procedure parameters (the **Number of parameters** field on the **Create Procedure (ADB26CO)** panel). This panel is displayed for each parameter that you need to define.

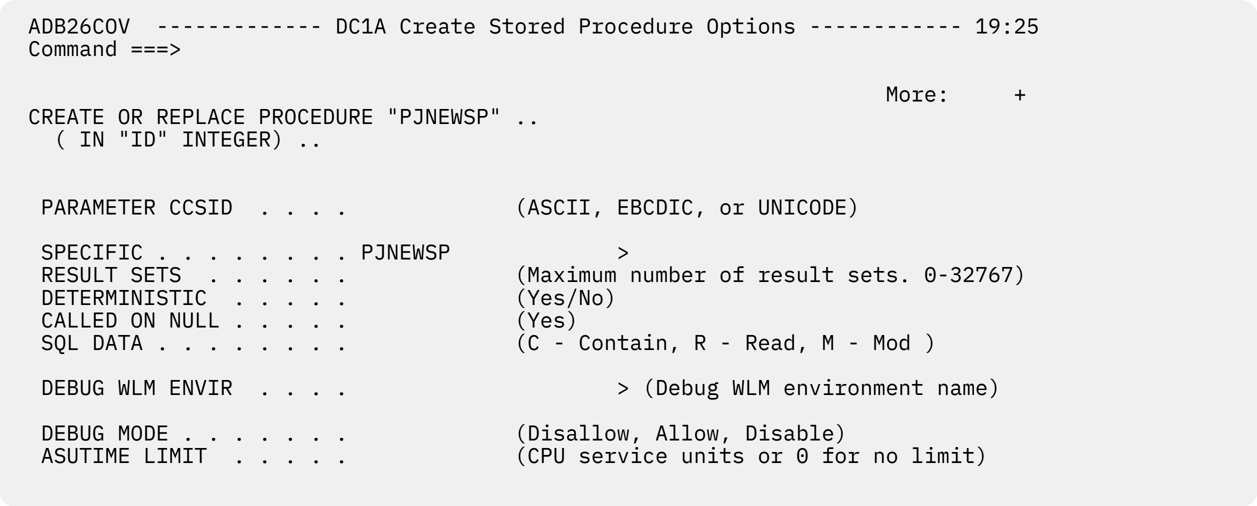


```
ADB26COU  ----------- DC1A Create Stored Procedure Parameters ----------- 19:05
Command ===>
                                                                   More:     +
CREATE OR REPLACE PROCEDURE "PJNEWSP" ..
 (parameter number 1)

 Parm type  . . . .                         (IN, OUT, or INOUT)
 Parm name  . . . .                      >  (Parameter name)

 For a non table like parameter specify:

 Data type  . . . .                         (Built-in only)
 Length . . . . . .                         (1 if DBCLOB with units indicator G)
 Scale  . . . . . .                         (used only w/DECIMAL and TIMESTAMP)
 WITH TIME ZONE . .                         (Yes/No - for TIMESTAMP only)

 FOR ? DATA . . . .                         (BIT, SBCS, or MIXED)
 CCSID  . . . . . .                         (ASCII, EBCDIC, or UNICODE)
 AS LOCATOR . . . .                         (Yes/No - for non-SQL only)
```

*Figure 560. **Create Stored Procedure Parameters (ADB26COU)** panel*

4. On the **Create Stored Procedure Options (ADB26COV)** panel, enter values for the relevant options, and press Enter.

   The format of the Create Stored Procedure Options (ADB26COV) panel varies depending on the language and type of stored procedure, the version of Db2, and the current APPLCOMPAT value.



```
ADB26COV  ------------- DC1A Create Stored Procedure Options ------------ 19:25
Command ===>
                                                                   More:     +
CREATE OR REPLACE PROCEDURE "PJNEWSP" ..
  ( IN "ID" INTEGER) ..


 PARAMETER CCSID  . . . .             (ASCII, EBCDIC, or UNICODE)

 SPECIFIC . . . . . . . . PJNEWSP             >
 RESULT SETS  . . . . . .             (Maximum number of result sets. 0-32767)
 DETERMINISTIC  . . . . .             (Yes/No)
 CALLED ON NULL . . . . .             (Yes)
 SQL DATA . . . . . . . .             (C - Contain, R - Read, M - Mod )

 DEBUG WLM ENVIR  . . . .                     > (Debug WLM environment name)

 DEBUG MODE . . . . . . .             (Disallow, Allow, Disable)
 ASUTIME LIMIT  . . . . .             (CPU service units or 0 for no limit)
```

*Figure 561. Example of the **Create Stored Procedure Options (ADB26COV)** panel*

5. Follow the remaining prompts to create the stored procedure.

The prompts vary depending on the type of procedure that you are creating and your Db2 Admin Tool settings. For example, if you are creating a native SQL procedure or an external SQL procedure, you are prompted to enter the procedure body. Also, if Change Management is enabled or the statement execution prompt is turned on, additional prompts are displayed.

When you complete all of the prompts, Db2 Admin Tool issues the SQL CREATE PROCEDURE statement with the parameters that you specify.

Scenario: Creating native SQL procedures

A *native SQL procedure* is a procedure whose body is written entirely in SQL and is created by issuing a single SQL statement, CREATE PROCEDURE.

Before you begin

Ensure that the **Manage Stored Procedures (ADB2ZP)** panel is open. For instructions on how to navigate to this panel, see "Stored procedures" on page 1013.

About this task

You can create a native SQL procedure to help with commonly performed tasks, such as creating a set of test objects. After you create a native SQL procedure in Db2 Admin Tool, you can then generate the DDL for that procedure and use that DDL to recreate that procedure in another database.

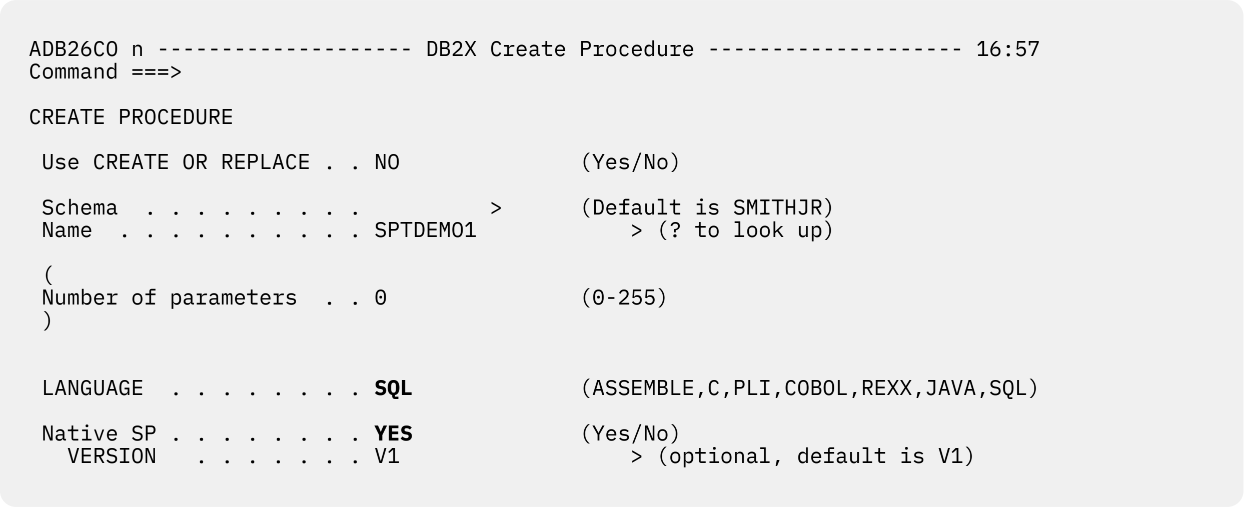
This scenario takes you through the following steps:

- Create a native SQL procedure.
- Generate the DDL for the procedure with masking into a work statement list.
- Run the work statement list to create the procedure in a different database.

Procedure

1. On the **Manage Stored Procedures (ADB2ZP)** panel, specify option 2, and press Enter.

   The **Create Procedure (ADB26CO)** panel is displayed. Alternatively, you can navigate to the **Create Procedure (ADB26CO)** panel by specifying option `2.4` on the **DB2 Administration Menu (ADB2)** panel and then specifying option `CO`.
2. On the **Create Procedure (ADB26CO)** panel, specify the required parameters, and press Enter:

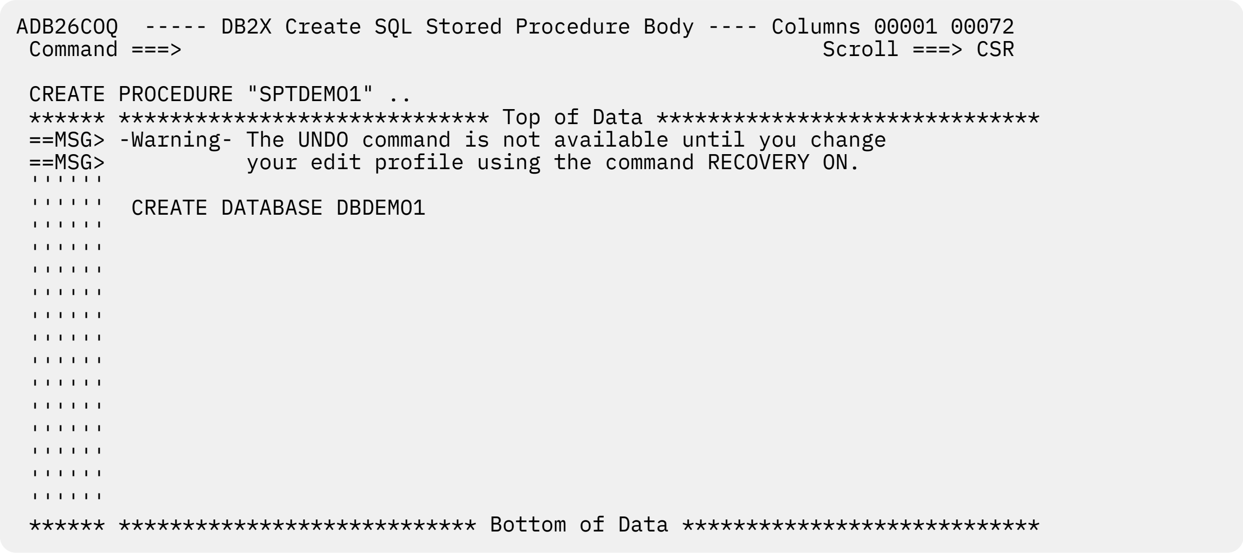


```
ADB26CO n -------------------- DB2X Create Procedure -------------------- 16:57
Command ===>

CREATE PROCEDURE

 Use CREATE OR REPLACE . . NO              (Yes/No)

 Schema  . . . . . . . . .          >      (Default is SMITHJR)
 Name  . . . . . . . . . . SPTDEMO01           > (? to look up)

 (
 Number of parameters  . . 0               (0-255)
 )


 LANGUAGE  . . . . . . . . SQL             (ASSEMBLE,C,PLI,COBOL,REXX,JAVA,SQL)

 Native SP . . . . . . . . YES             (Yes/No)
   VERSION   . . . . . . . V1                  > (optional, default is V1)
```

*Figure 562. **Create Procedure (ADB26CO)** panel*

   For a native SQL procedure, specify SQL in the **LANGUAGE** field and Y in the **Native SP** field.
3. On the **Create Stored Procedure Options (ADB26COV)** panel, enter any desired parameter values, and press Enter.

4. On the **Create SQL Stored Procedure Body (ADB26COQ)** panel, enter the SQL procedure body, as shown in the following example:

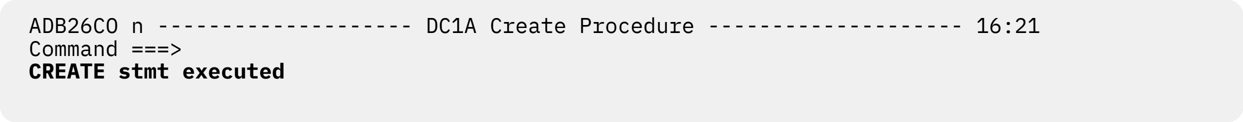


```
ADB26COQ  ----- DB2X Create SQL Stored Procedure Body ---- Columns 00001 00072
 Command ===>                                                  Scroll ===> CSR

 CREATE PROCEDURE "SPTDEMO1" ..
 ****** ***************************** Top of Data ******************************
 ==MSG> -Warning- The UNDO command is not available until you change
 ==MSG>           your edit profile using the command RECOVERY ON.
 ''''''
 ''''''  CREATE DATABASE DBDEMO01
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ''''''
 ****** **************************** Bottom of Data ****************************
```

*Figure 563. **Create SQL Stored Procedure Body (ADB26COQ)** panel*

5. Exit the **Create SQL Stored Procedure Body (ADB26COQ)** panel.

   The following message is displayed to confirm that the procedure is created:

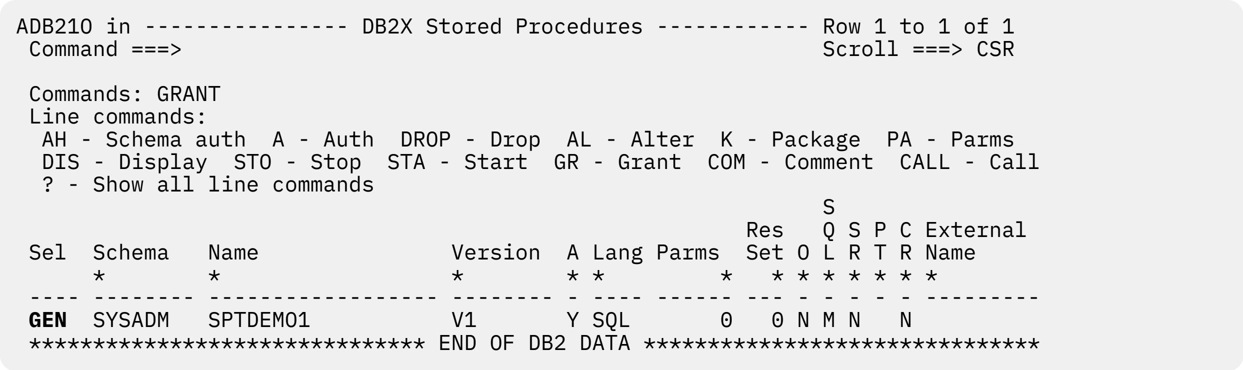


```
 ADB26CO n -------------------- DC1A Create Procedure -------------------- 16:21
 Command ===>
 CREATE stmt executed
```

   Note: Depending on your Db2 Admin Tool settings, you might first see the **Statement Execution Prompt (ADB2PSTM)** panel, the **Change Management Prompt (ADB2CMPR)** panel, or both.

   Now, you can generate the DDL for the procedure by using either the DDL line command or the GEN function. The next steps show you how to use the GEN function to generate the DDL with masking into a work statement list.

6. Exit back to the **DB2 Administration Menu (ADB2)** panel.
7. Specify option 1, and press Enter.
8. On the **System Catalog (ADB21)** panel, specify option O, and press Enter.
9. On the **Stored Procedures (ADB21O)** panel, specify GEN next to your SQL procedure name, and press Enter:



```
ADB21O in ---------------- DB2X Stored Procedures ------------ Row 1 to 1 of 1
 Command ===>                                                  Scroll ===> CSR

 Commands: GRANT
 Line commands:
  AH - Schema auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
  DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment  CALL - Call
  ? - Show all line commands
                                                               S
                                                         Res   Q S P C External
 Sel  Schema   Name               Version  A Lang Parms  Set O L R T R Name
      *        *                  *        * *         *   * * * * * * *
 ---- -------- ------------------ -------- - ---- ------ --- - - - - - ---------
 GEN  SYSADM   SPTDEMO1           V1       Y SQL       0   0 N M N   N
 ******************************* END OF DB2 DATA *******************************
```

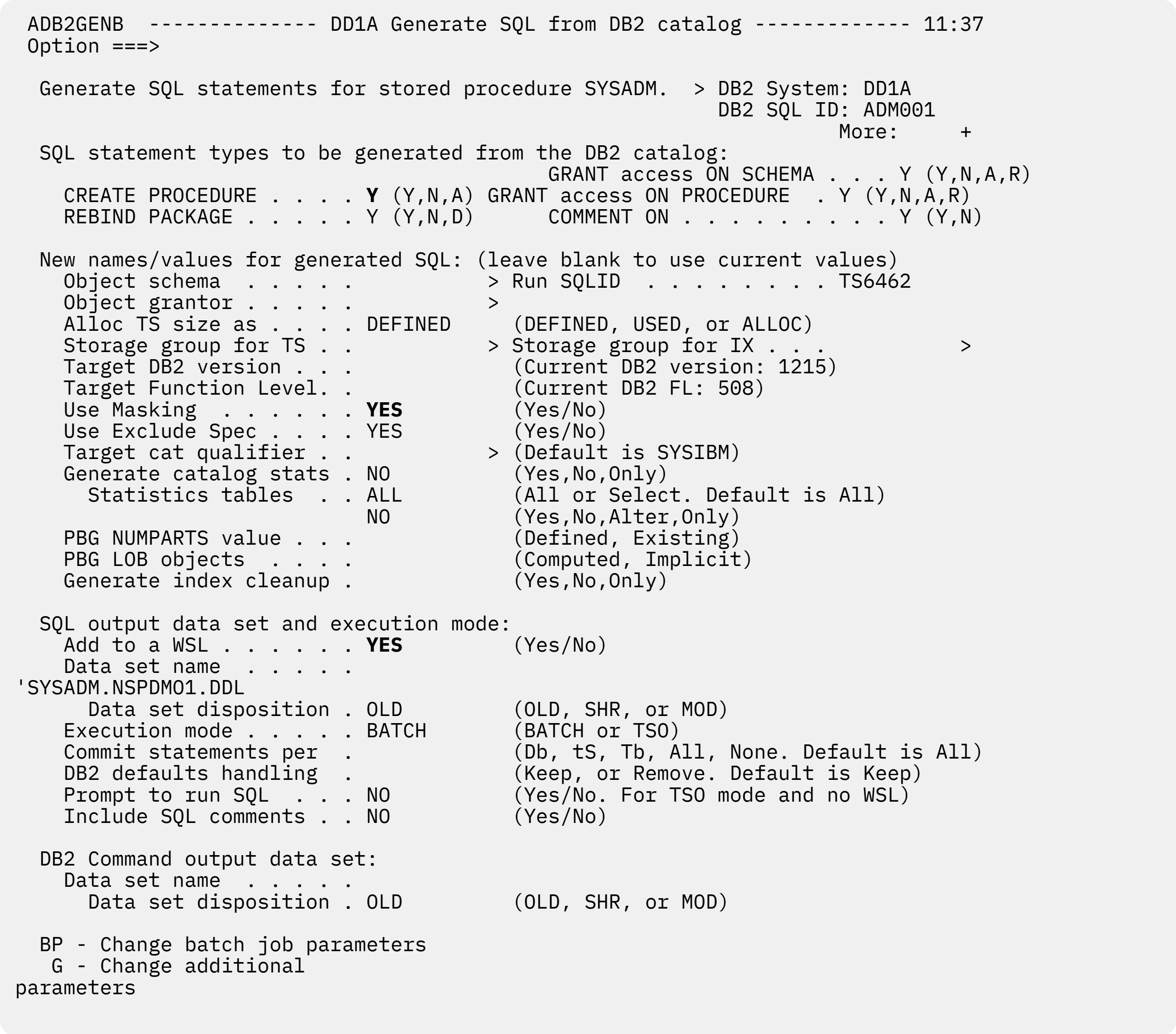
*Figure 564. **Stored Procedures (ADB21O)** panel*

10. On the **Generate SQL from DB2 catalog (ADB2GENB)** panel, specify the following options to generate the DDL with masking into a work statement list (WSL):

CREATE PROCEDURE
Y

Use Masking
YES

Add to work stmt list
YES



```
 ADB2GENB  -------------- DD1A Generate SQL from DB2 catalog ------------- 11:37
 Option ===>

  Generate SQL statements for stored procedure SYSADM.  > DB2 System: DD1A
                                                          DB2 SQL ID: ADM001
                                                                    More:     +
  SQL statement types to be generated from the DB2 catalog:
                                            GRANT access ON SCHEMA . . . Y (Y,N,A,R)
    CREATE PROCEDURE . . . . Y (Y,N,A) GRANT access ON PROCEDURE  . Y (Y,N,A,R)
    REBIND PACKAGE . . . . . Y (Y,N,D)      COMMENT ON . . . . . . . . . Y (Y,N)

  New names/values for generated SQL: (leave blank to use current values)
    Object schema  . . . . .           > Run SQLID  . . . . . . . . TS6462
    Object grantor . . . . .           >
    Alloc TS size as . . . . DEFINED     (DEFINED, USED, or ALLOC)
    Storage group for TS . .           > Storage group for IX . . .           >
    Target DB2 version . . .             (Current DB2 version: 1215)
    Target Function Level. .             (Current DB2 FL: 508)
    Use Masking  . . . . . . YES         (Yes/No)
    Use Exclude Spec . . . . YES         (Yes/No)
    Target cat qualifier . .           > (Default is SYSIBM)
    Generate catalog stats . NO          (Yes,No,Only)
      Statistics tables  . . ALL         (All or Select. Default is All)
                             NO          (Yes,No,Alter,Only)
    PBG NUMPARTS value . . .             (Defined, Existing)
    PBG LOB objects  . . . .             (Computed, Implicit)
    Generate index cleanup .             (Yes,No,Only)

  SQL output data set and execution mode:
    Add to a WSL . . . . . . YES         (Yes/No)
    Data set name  . . . . .
'SYSADM.NSPDMO1.DDL
      Data set disposition . OLD         (OLD, SHR, or MOD)
    Execution mode . . . . . BATCH       (BATCH or TSO)
    Commit statements per  .             (Db, tS, Tb, All, None. Default is All)
    DB2 defaults handling  .             (Keep, or Remove. Default is Keep)
    Prompt to run SQL  . . . NO          (Yes/No. For TSO mode and no WSL)
    Include SQL comments . . NO          (Yes/No)

  DB2 Command output data set:
    Data set name  . . . . .
      Data set disposition . OLD         (OLD, SHR, or MOD)

  BP - Change batch job parameters
   G - Change additional
parameters
```

*Figure 565. **Generate SQL from DB2 catalog (ADB2GENB)** panel*

Note: The R option for **CREATE PROCEDURE** is displayed only if you are running Db2 12 and the current APPLCOMPAT value is 507 or later.

11. Specify any other desired options, and press Enter.
12. On the **Specify Masks (ADB2GENM)** panel, specify the appropriate masks.

    In this case, you want to create a mask for the database name and stored procedure name so that when you run the DDL, the stored procedure will be created in a different database and have a different name. For example, you can specify `SPTDEM01, SPTDEM02` for **STPNAME** and `DBDEM01, DBDEM02` for **DBNAME**.

    For help specifying masks, see "Specifying masks" on page 315.
13. On the **Specify Work Statement List (ADB2WLDA)** panel, specify the WSL to which you want to add this DDL, and press Enter:

```
EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e ADB2WLDA  ------------------ Specify Work Statement List ------------------ e
e                                                                             e
e                                                                             e
e  Work stmt list dsn  . . . 'SYSADM.NSPDEMO2.WSL'                            e
e  Work stmt list name . . . NSPDEMO2                                         e
e                                                                             e
e                                                                             e
e                                                                             e
e                                                                             e
e                                                                             e
e                                                                             e
e                                                                             e
e                                                                             e
DsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 566. ***Specify Work Statement List (ADB2WLDA)*** *panel*

The following message confirms that the DDL was added to the WSL:

```
ADB21O in                   DC1A Stored Procedures             Row 1 to 1 of 1
Statements added
```

14. Exit back to the **DB2 Administration Menu (ADB2)** panel.
15. Specify `W`, and press Enter.
16. On the **Manage Work Statement Lists (ADB2W)** panel, specify option 1, and press Enter:

```
ADB2W min --------------- DD1A Manage Work Statement Lists -------------- 15:43
 Option ===> 1

    1 - Show work statement list library                  DB2 System: DD1A
    2 - Show work statement list                          DB2 SQL ID: ADM001

  Work stmt list dsn  ===> 'SYSADM.NSPDEMO2.WSL'
  Work stmt list name ===> NSPDEMO2
```

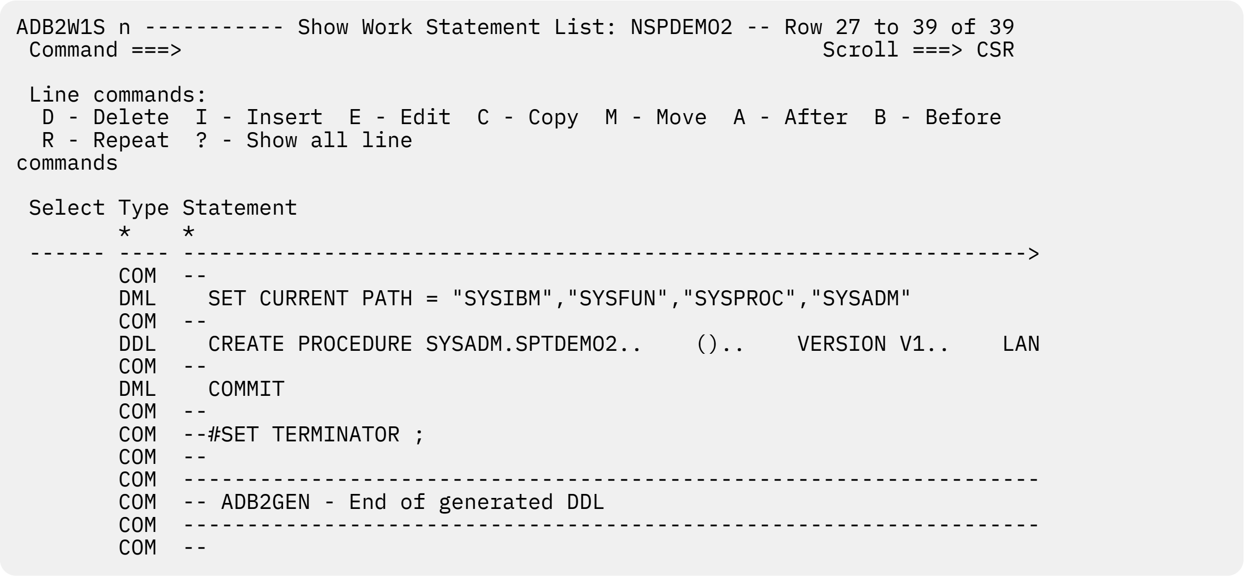
Figure 567. ***Manage Work Statement Lists (ADB2W)*** *panel*

17. On the **Work Statement List Library (ADB2W1)** panel, specify the S line command next to your work statement list:

```
ADB2W1 in ----------------- Work Statement List Library ------ Row 1 to 1 of 1
 Command ===>                                                  Scroll ===> CSR

 Line commands:
  S - Show  R - Run (batch)   D - Delete  C - Copy  A - Append  Q - Clone
  I - Interpret  V - Validate  E - Edit  B - Checkpoint
  ? - Show all line commands

 Work Statement List: 'SYSADM.NSPDEMO2.WSL'

 Sel Name     Created    Changed          ID
Restart
     *        *          *                *        *
 --- -------- ---------- ---------------- -------- --------
  S  NSPDEMO2 2009/06/01 2009/06/01 15:43 SYSADM
 ******************************* END OF DB2 DATA *******************************
```

Figure 568. ***Work Statement List Library (ADB2W1)*** *panel*

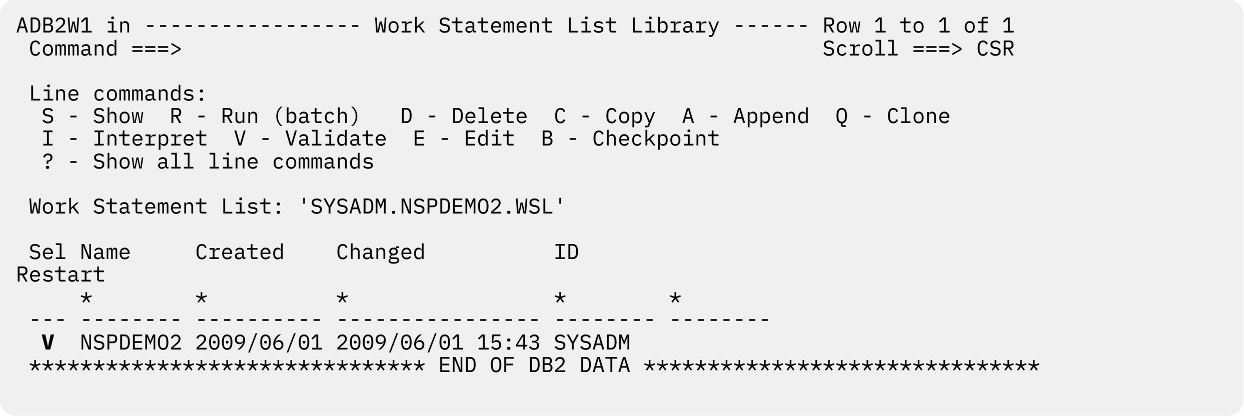
The **Show Work Statement List (ADB2W1S)** panel is displayed. You can scroll through this WSL to see your DDL:



```
ADB2W1S n ----------- Show Work Statement List: NSPDEMO2 -- Row 27 to 39 of 39
 Command ===>                                                  Scroll ===> CSR

 Line commands:
  D - Delete  I - Insert  E - Edit  C - Copy  M - Move  A - After  B - Before
  R - Repeat  ? - Show all line
commands

 Select Type Statement
        *    *
 ------ ---- ------------------------------------------------------------------>
        COM  --
        DML    SET CURRENT PATH = "SYSIBM","SYSFUN","SYSPROC","SYSADM"
        COM  --
        DDL    CREATE PROCEDURE SYSADM.SPTDEMO2..    ()..    VERSION V1..    LAN
        COM  --
        DML    COMMIT
        COM  --
        COM  --#SET TERMINATOR ;
        COM  --
        COM  -------------------------------------------------------------------
        COM  -- ADB2GEN - End of generated DDL
        COM  -------------------------------------------------------------------
        COM  --
```

*Figure 569. **Show Work Statement List (ADB2W1S)** panel*

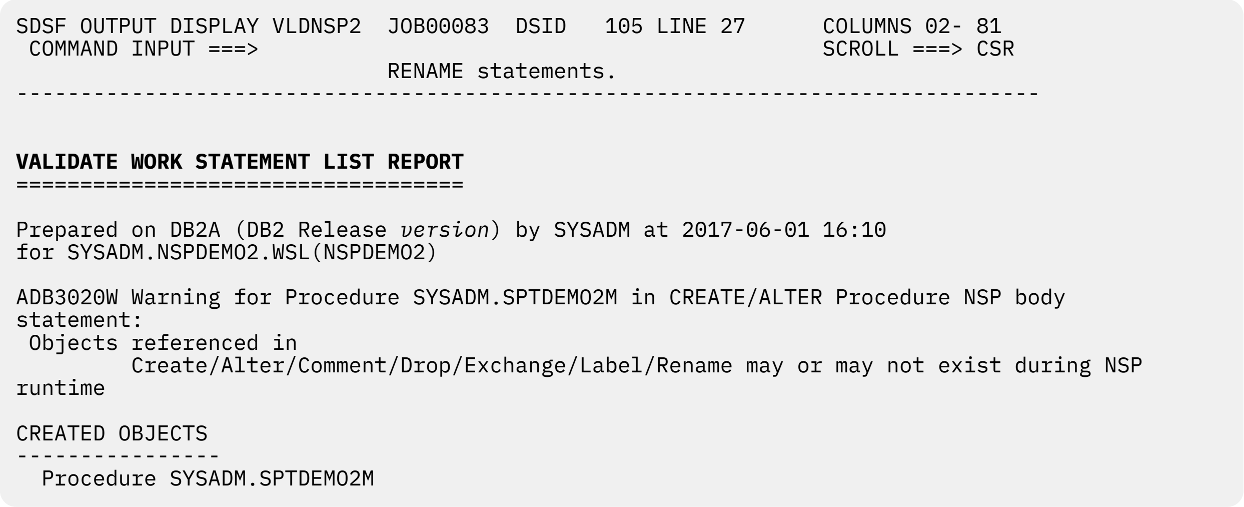
18. Exit back to the **Work Statement List Library (ADB2W1)** panel, and specify the V line command next to your work statement list, and press Enter:



```
ADB2W1 in ----------------- Work Statement List Library ------ Row 1 to 1 of 1
 Command ===>                                                  Scroll ===> CSR

 Line commands:
  S - Show  R - Run (batch)   D - Delete  C - Copy  A - Append  Q - Clone
  I - Interpret  V - Validate  E - Edit  B - Checkpoint
  ? - Show all line commands

 Work Statement List: 'SYSADM.NSPDEMO2.WSL'

 Sel Name     Created    Changed          ID
Restart
     *        *          *                *        *
 --- -------- ---------- ---------------- -------- --------
  V  NSPDEMO2 2009/06/01 2009/06/01 15:43 SYSADM
 ******************************* END OF DB2 DATA *******************************
```

Figure 570. Work Statement List Library panel

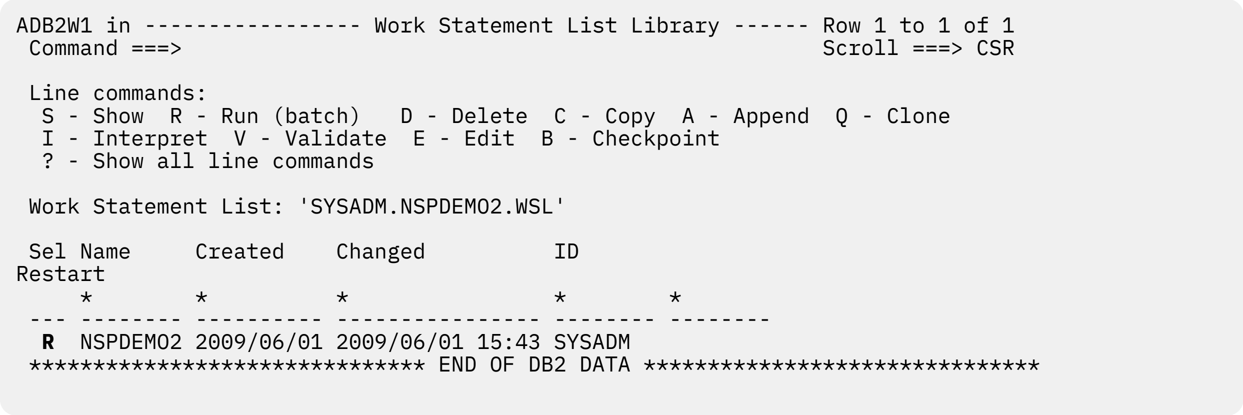
19. Submit the generated JCL and view the output, as shown in the following example:



```
SDSF OUTPUT DISPLAY VLDNSP2  JOB00083  DSID   105 LINE 27      COLUMNS 02- 81
 COMMAND INPUT ===>                                            SCROLL ===> CSR
                             RENAME statements.
--------------------------------------------------------------------------------


VALIDATE WORK STATEMENT LIST REPORT
===================================

Prepared on DB2A (DB2 Release version) by SYSADM at 2017-06-01 16:10
for SYSADM.NSPDEMO2.WSL(NSPDEMO2)

ADB3020W Warning for Procedure SYSADM.SPTDEMO2M in CREATE/ALTER Procedure NSP body
statement:
 Objects referenced in
        Create/Alter/Comment/Drop/Exchange/Label/Rename may or may not exist during NSP
runtime

CREATED OBJECTS
----------------
  Procedure SYSADM.SPTDEMO2M
```

Figure 571. Validation Work Statement List Report

20. After you validate the work statement list, return to the **Work Statement List Library (ADB2W1)** panel and specify the R line command to run the WSL, and press Enter:

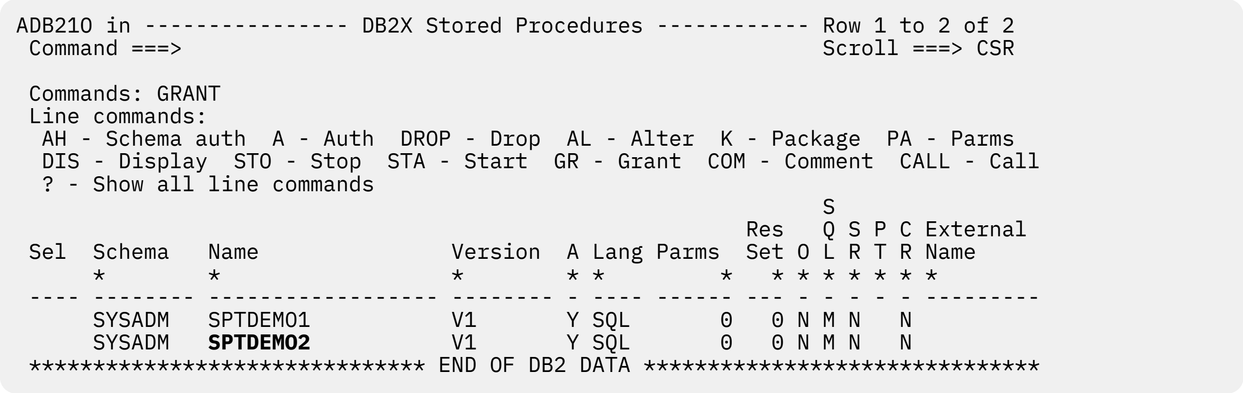


```
ADB2W1 in ----------------- Work Statement List Library ------ Row 1 to 1 of 1
 Command ===>                                                  Scroll ===> CSR

 Line commands:
  S - Show  R - Run (batch)   D - Delete  C - Copy  A - Append  Q - Clone
  I - Interpret  V - Validate  E - Edit  B - Checkpoint
  ? - Show all line commands

 Work Statement List: 'SYSADM.NSPDEMO2.WSL'

 Sel Name     Created    Changed          ID
Restart
     *        *          *                *        *
 --- -------- ---------- ---------------- -------- --------
  R  NSPDEMO2 2009/06/01 2009/06/01 15:43 SYSADM
 ******************************* END OF DB2 DATA *******************************
```

*Figure 572. **Work Statement List Library (ADB2W1)** panel*

21. Submit the generated JCL and confirm that the job ran successfully.
22. Return to panel the **Stored Procedures (ADB21O)** panel and verify that the SPTDEMO2 native SQL procedure was created successfully:

    Note: To navigate to this panel, from **DB2 Administration Menu (ADB2)** panel, specify option 1 and then 0.



```
ADB21O in ---------------- DB2X Stored Procedures ------------ Row 1 to 2 of 2
 Command ===>                                                  Scroll ===> CSR

 Commands: GRANT
 Line commands:
  AH - Schema auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
  DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment  CALL - Call
  ? - Show all line commands
                                                               S
                                                         Res   Q S P C External
 Sel  Schema   Name               Version  A Lang Parms  Set O L R T R Name
      *        *                  *        * *         *   * * * * * * *
 ---- -------- ------------------ -------- - ---- ------ --- - - - - - ---------
      SYSADM   SPTDEMO1           V1       Y SQL       0   0 N M N   N
      SYSADM   SPTDEMO2           V1       Y SQL       0   0 N M N   N
 ******************************* END OF DB2 DATA *******************************
```

*Figure 573. **Stored Procedures (ADB21O)** panel*

Calling stored procedures

You can call a stored procedure directly from the panel interface in Db2 Admin Tool.

Before you begin

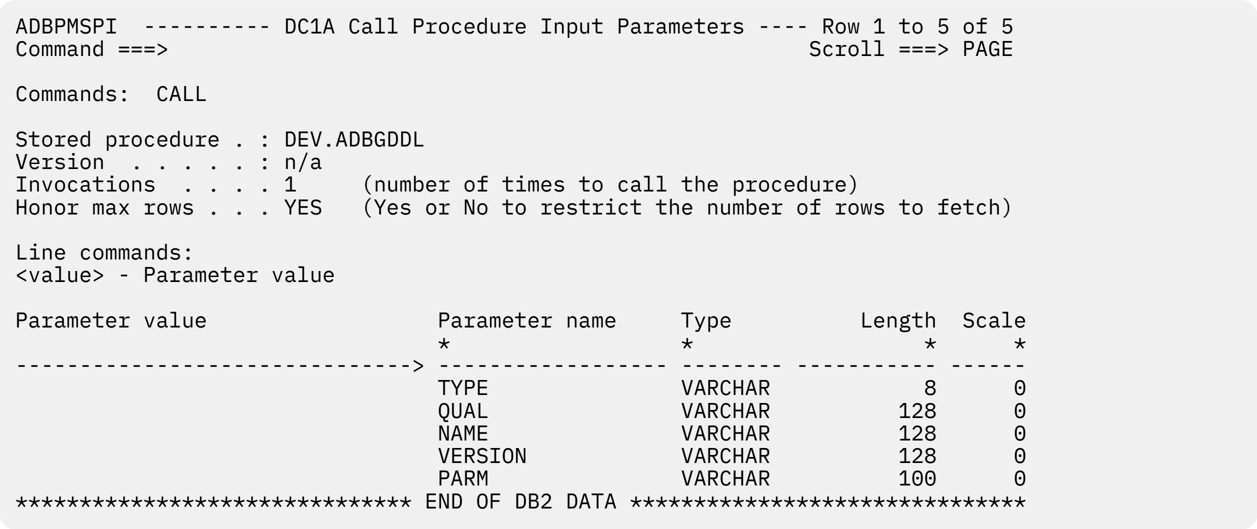
Ensure that the **Manage Stored Procedures (ADB2ZP)** panel is open. For instructions on how to navigate to this panel, see "Stored procedures" on page 1013.

Procedure

To call a stored procedure:

1. On the **Manage Stored Procedures (ADB2ZP)** panel, specify option 1, and press Enter.
2. On the **Stored Procedures (ADB21O)** panel, specify the CALL line command next to the procedure that you want to call, and press Enter.
3. If the **Call Procedure Input Parameters (ADBPMSPI)** panel is displayed, specify any input parameter values in the **Parameter value** column, specify the CALL command, and press Enter.

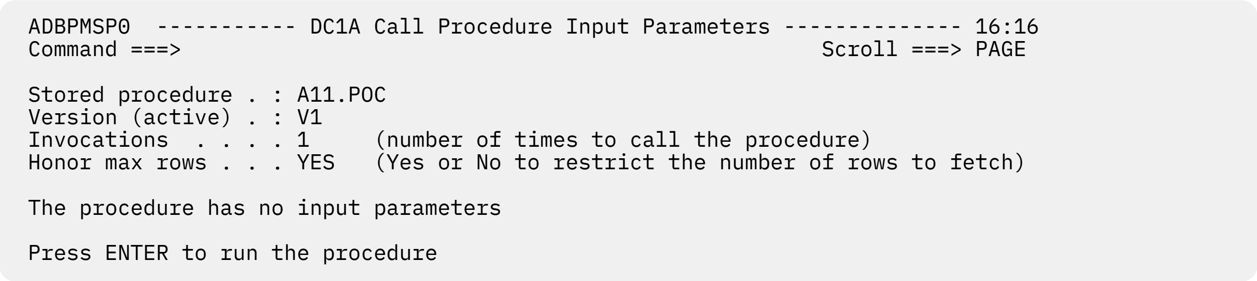
Restriction: If you are calling ADBGDDL, any QUAL and NAME values that start with a blank must be surrounded with double quotation marks. Any values that start with a blank cannot be more than 127 characters long.



```
ADBPMSPI  ---------- DC1A Call Procedure Input Parameters ---- Row 1 to 5 of 5
Command ===>                                                  Scroll ===> PAGE

Commands:  CALL

Stored procedure . : DEV.ADBGDDL
Version  . . . . . : n/a
Invocations  . . . . 1     (number of times to call the procedure)
Honor max rows . . . YES   (Yes or No to restrict the number of rows to fetch)

Line commands:
<value> - Parameter value

Parameter value                  Parameter name     Type          Length  Scale
                                 *                  *                  *      *
------------------------------>  ------------------ -------- ----------- ------
                                 TYPE               VARCHAR            8      0
                                 QUAL               VARCHAR          128      0
                                 NAME               VARCHAR          128      0
                                 VERSION            VARCHAR          128      0
                                 PARM               VARCHAR          100      0
******************************* END OF DB2 DATA *******************************
```

Figure 574. ***Call Procedure Input Parameters (ADBPMSPI)*** *panel*

4. If the **Call Procedure Input Parameters (ADBPMSP0)** panel opens to show that the procedure does not have any input parameters, press Enter to run the procedure:



```
ADBPMSP0  ----------- DC1A Call Procedure Input Parameters ------------- 16:16
Command ===>                                                  Scroll ===> PAGE

Stored procedure . : A11.POC
Version (active) . : V1
Invocations  . . . . 1     (number of times to call the procedure)
Honor max rows . . . YES   (Yes or No to restrict the number of rows to fetch)

The procedure has no input parameters

Press ENTER to run the procedure
```

5. On the **Call Procedure Results (ADBPMSPO)** panel, examine the results. Use the S line command to view any result sets.

Displaying or altering stored procedures

You can display all the stored procedures that you have defined in your system and make changes to any of them as needed.

Before you begin

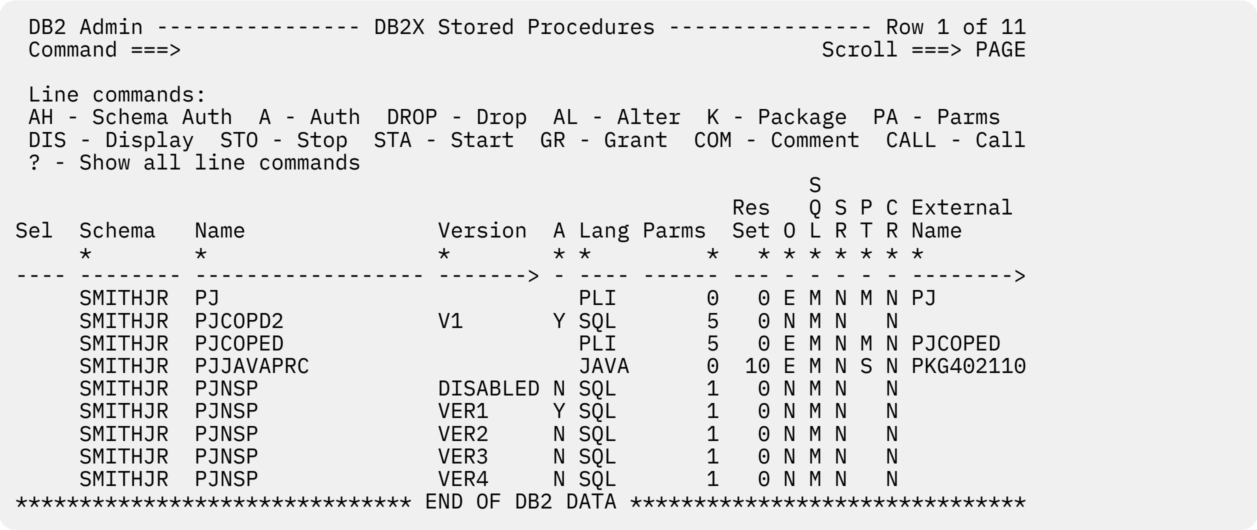
Ensure that the **Manage Stored Procedures (ADB2ZP)** panel is open. For instructions on how to navigate to this panel, see "Stored procedures" on page 1013.

Procedure

To display stored procedures:

1. On the **Manage Stored Procedures (ADB2ZP)** panel, specify option 1, and press Enter.

   The **Stored Procedures (ADB21O)** panel lists the stored procedures that are defined on your system:



```
 DB2 Admin ---------------- DB2X Stored Procedures ---------------- Row 1 of 11
 Command ===>                                                  Scroll ===> PAGE

 Line commands:
 AH - Schema Auth  A - Auth  DROP - Drop  AL - Alter  K - Package  PA - Parms
 DIS - Display  STO - Stop  STA - Start  GR - Grant  COM - Comment  CALL - Call
 ? - Show all line commands
                                                              S
                                                        Res   Q S P C External
Sel  Schema   Name               Version  A Lang Parms  Set O L R T R Name
     *        *                  *        * *         *   * * * * * * *
---- -------- ------------------ -------> - ---- ------ --- - - - - - -------->
     SMITHJR  PJ                            PLI       0   0 E M N M N PJ
     SMITHJR  PJCOPD2            V1       Y SQL       5   0 N M N   N
     SMITHJR  PJCOPED                       PLI       5   0 E M N M N PJCOPED
     SMITHJR  PJJAVAPRC                     JAVA      0  10 E M N S N PKG402110
     SMITHJR  PJNSP              DISABLED N SQL       1   0 N M N   N
     SMITHJR  PJNSP              VER1     Y SQL       1   0 N M N   N
     SMITHJR  PJNSP              VER2     N SQL       1   0 N M N   N
     SMITHJR  PJNSP              VER3     N SQL       1   0 N M N   N
     SMITHJR  PJNSP              VER4     N SQL       1   0 N M N   N
******************************* END OF DB2 DATA *******************************
```

*Figure 575. **Stored Procedures (ADB21O)** panel*

From this panel, you can use the line commands to perform actions on individual stored procedures.

Restriction: The SRC line command is not supported for native SQL procedures. Press PF1 if you get an invalid line command message and look at the **O** column. If there is an **N** in that column, then the SRC command is not supported. An E in the O column indicates the SRC command is supported.

2. If you need to alter a stored procedure, use the ALT or AL line commands to make changes.

Related tasks

“Creating views of SYSIBM.SYSROUTINES” on page 1023
SYSIBM.SYSROUTINES is the Db2 catalog table that stores information about all routines, including stored procedures. You can create a view on SYSROUTINES to enable other people to administer their own stored procedures.

“Displaying stored procedure statistics” on page 1025
You can get information about stored procedures that are accessed by Db2 applications, such as their current status and the number of requests that are currently running and queued.

Creating views of SYSIBM.SYSROUTINES

SYSIBM.SYSROUTINES is the Db2 catalog table that stores information about all routines, including stored procedures. You can create a view on SYSROUTINES to enable other people to administer their own stored procedures.

Before you begin

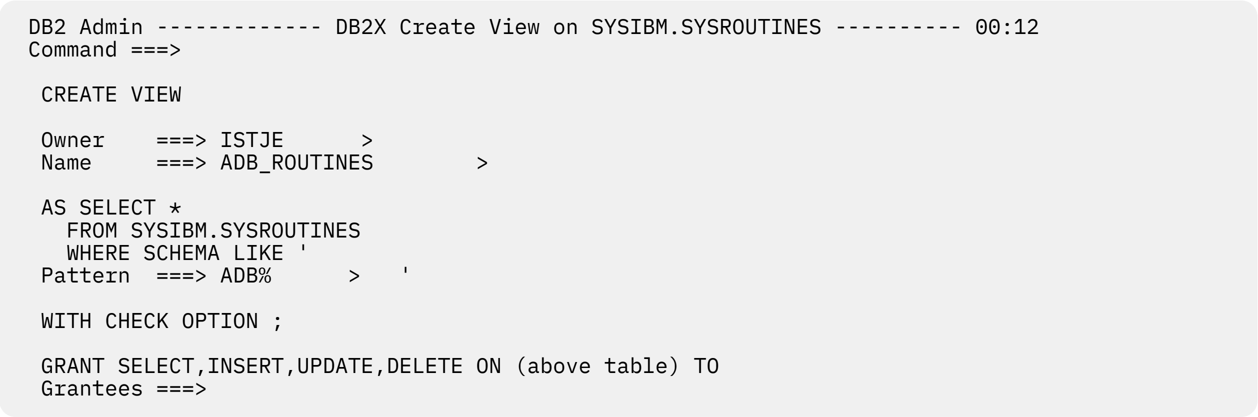
Ensure that the **Manage Stored Procedures (ADB2ZP)** panel is open. For instructions on how to navigate to this panel, see “Stored procedures” on page 1013.

Procedure

To create a view of SYSIBM.SYSROUTINES:

1. On the **Manage Stored Procedures (ADB2ZP)** panel, specify option 6, and press Enter.

   The **Create View on SYSIBM.SYSROUTINES (ADB2ZF6)** panel is displayed, as shown in the following figure. This panel lets you define a view for all procedures with the (LIKE) pattern you define.



```
DB2 Admin ------------- DB2X Create View on SYSIBM.SYSROUTINES ---------- 00:12
Command ===>

 CREATE VIEW

 Owner    ===> ISTJE      >
 Name     ===> ADB_ROUTINES        >

 AS SELECT *
   FROM SYSIBM.SYSROUTINES
   WHERE SCHEMA LIKE '
 Pattern  ===> ADB%      >   '

 WITH CHECK OPTION ;

 GRANT SELECT,INSERT,UPDATE,DELETE ON (above table) TO
 Grantees ===>
```

*Figure 576. **Create View on SYSIBM.SYSROUTINES (ADB2ZF6)** panel*

2. Fill in the fields on this panel to create a view. For example, define view ABC.PROCEDURES as a view on SYSIBM.SYSROUTINES WHERE SCHEMA LIKE 'ABC.

   View ABC.PROCEDURES contains all stored procedures with the schema starting with ABC. In addition, you can issue GRANT SELECT, INSERT, UPDATE, or DELETE statements on the view to a list of authorization IDs (grantees).

Related tasks

“Displaying views on SYSIBM.SYSROUTINES” on page 1024
You can display a list of all views that are defined on SYSIBM.SYSROUTINES.

“Displaying or altering stored procedures ” on page 1022
You can display all the stored procedures that you have defined in your system and make changes to any of them as needed.

Related information

SYSROUTINES catalog table (Db2 12 for z/OS documentation)

Displaying views on SYSIBM.SYSROUTINES

You can display a list of all views that are defined on SYSIBM.SYSROUTINES.

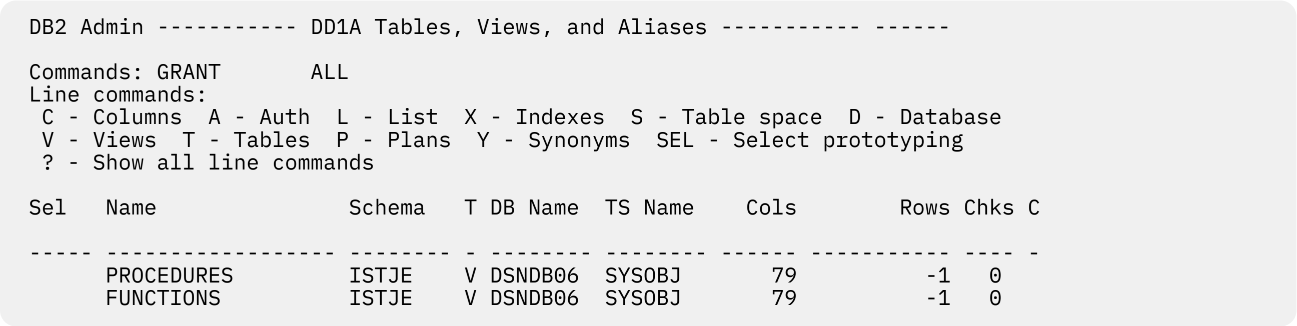
Before you begin

Ensure that the **Manage Stored Procedures (ADB2ZP)** panel is open. For instructions on how to navigate to this panel, see “Stored procedures” on page 1013.

Procedure

On the **Manage Stored Procedures (ADB2ZP)** panel, specify option 7, and press Enter.

The **Tables, Views, and Aliases (ADB21T)** panel is displayed, as shown in the following figure. This panel shows the views that exist on SYSIBM.SYSROUTINES. That is, it shows the views that were created by using option 6 on the **Manage Stored Procedures (ADB2ZP)** panel.



```
DB2 Admin ----------- DD1A Tables, Views, and Aliases ----------- ------

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

Sel   Name               Schema   T DB Name  TS Name    Cols        Rows Chks C

----- ------------------ -------- - -------- -------- ------ ----------- ---- -
      PROCEDURES         ISTJE    V DSNDB06  SYSOBJ       79          -1   0
      FUNCTIONS          ISTJE    V DSNDB06  SYSOBJ       79          -1   0
```

*Figure 577. **Tables, Views, and Aliases (ADB21T)** panel showing views on SYSIBM.SUBROUTINES (ADB21T)*

Related tasks

“Creating views of SYSIBM.SYSROUTINES” on page 1023
SYSIBM.SYSROUTINES is the Db2 catalog table that stores information about all routines, including stored procedures. You can create a view on SYSROUTINES to enable other people to administer their own stored procedures.

Displaying stored procedure statistics

You can get information about stored procedures that are accessed by Db2 applications, such as their current status and the number of requests that are currently running and queued.

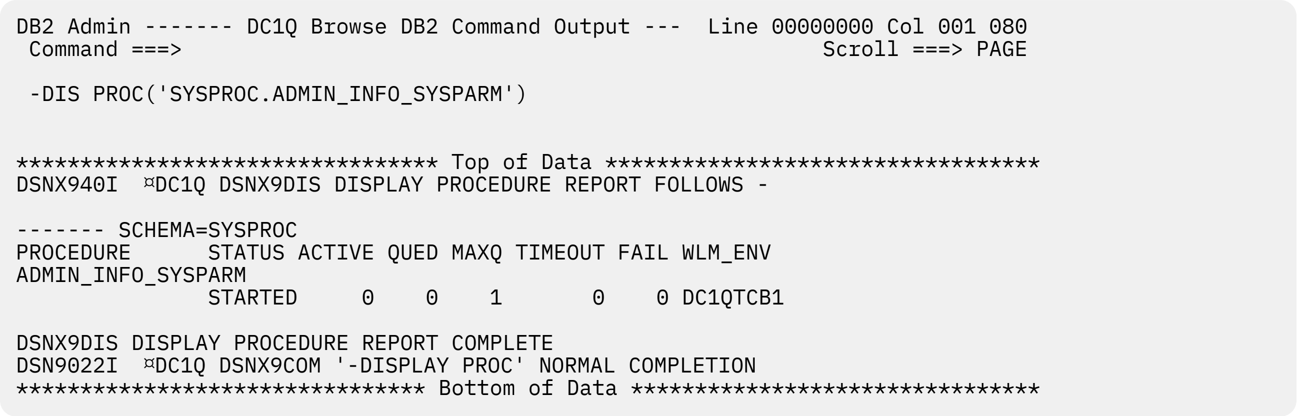
Before you begin

Ensure that the **Manage Stored Procedures (ADB2ZP)** panel is open. For instructions on how to navigate to this panel, see “Stored procedures” on page 1013.

Procedure

On the **Manage Stored Procedures (ADB2ZP)** panel, specify option 3, and press Enter.

The **Display Stored Procedure Statistics (ADB2DB2O)** panel is displayed. This panel shows statistics for stored procedures that are accessed by Db2 applications:



```
DB2 Admin ------- DC1Q Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

 -DIS PROC('SYSPROC.ADMIN_INFO_SYSPARM')


********************************* Top of Data **********************************
DSNX940I  ¤DC1Q DSNX9DIS DISPLAY PROCEDURE REPORT FOLLOWS -

------- SCHEMA=SYSPROC
PROCEDURE      STATUS ACTIVE QUED MAXQ TIMEOUT FAIL WLM_ENV
ADMIN_INFO_SYSPARM
               STARTED     0    0    1       0    0 DC1QTCB1

DSNX9DIS DISPLAY PROCEDURE REPORT COMPLETE
DSN9022I  ¤DC1Q DSNX9COM '-DISPLAY PROC' NORMAL COMPLETION
******************************** Bottom of Data ********************************
```

*Figure 578. **Display Stored Procedure Statistics (ADB2DB2O)** panel*

Related tasks

“Displaying or altering stored procedures ” on page 1022

You can display all the stored procedures that you have defined in your system and make changes to any of them as needed.

Starting or stopping all stored procedures

Starting stored procedures activates the definition of a stored procedure that is stopped or refreshes one that is stored in the cache. Conversely, stopping stored procedures prevents Db2 from accepting SQL CALL statements for one or more stored procedures.

You do not need to start procedures when you define a new stored procedure; Db2 automatically activates the new definition when it first receives an SQL CALL statement for the new procedure.

Before you begin

Ensure that the **Manage Stored Procedures (ADB2ZP)** panel is open. For instructions on how to navigate to this panel, see "Stored procedures" on page 1013.

Procedure

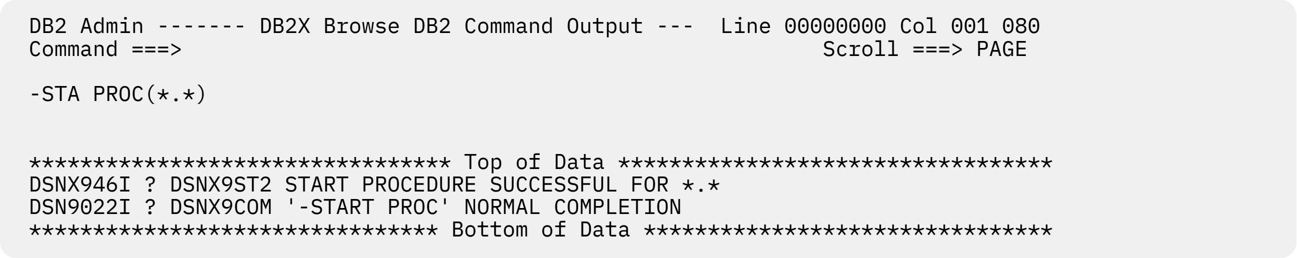
On the**Manage Stored Procedures (ADB2ZP)** panel, specify one of the following options, and press Enter

4
: Starts all stored procedures

5
: Stops all stored procedures

Db2 Admin Tool issues the Db2 START PROCEDURE(*.*) or STOP PROCEDURE (*.*) command as requested, and displays the status of the command.

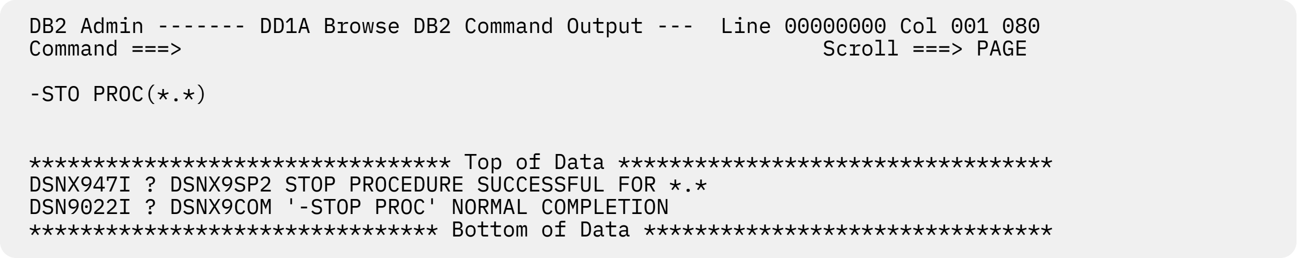
The following figure shows example output for START PROCEDURE:



```
DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                                Scroll ===> PAGE

-STA PROC(*.*)


********************************* Top of Data **********************************
DSNX946I ? DSNX9ST2 START PROCEDURE SUCCESSFUL FOR *.*
DSN9022I ? DSNX9COM '-START PROC' NORMAL COMPLETION
******************************** Bottom of Data ********************************
```

Figure 579. Start All Stored Procedures panel (ADB2DB2O)

The following figure shows example output for STOP PROCEDURE:



```
DB2 Admin ------- DD1A Browse DB2 Command Output ---  Line 00000000 Col 001 080
Command ===>                                                Scroll ===> PAGE

-STO PROC(*.*)


********************************* Top of Data **********************************
DSNX947I ? DSNX9SP2 STOP PROCEDURE SUCCESSFUL FOR *.*
DSN9022I ? DSNX9COM '-STOP PROC' NORMAL COMPLETION
******************************** Bottom of Data ********************************
```

Figure 580. Stop All Stored Procedures panel (ADB2DB2O)

Function management

Use the information in this section to learn how to use Db2 Admin Tool to manage functions.

Displaying or altering functions

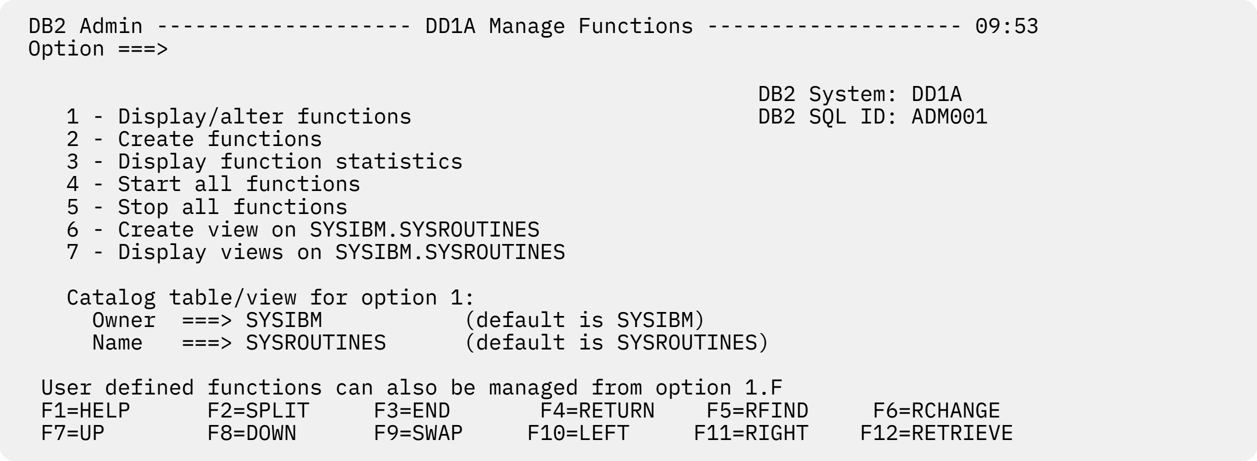
This procedure explains how to display or alter functions.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option FM, and press Enter.

   The **Manage Functions (ADB2ZF)** panel is displayed, as shown in the following figure.

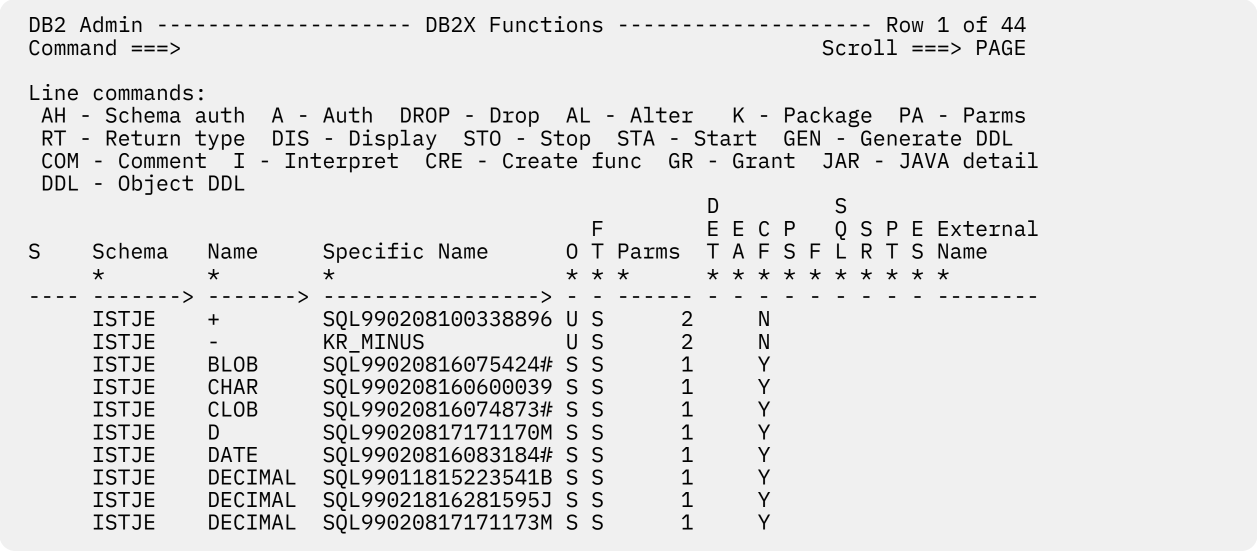


```
DB2 Admin -------------------- DD1A Manage Functions -------------------- 09:53
Option ===>

                                                         DB2 System: DD1A
   1 - Display/alter functions                           DB2 SQL ID: ADM001
   2 - Create functions
   3 - Display function statistics
   4 - Start all functions
   5 - Stop all functions
   6 - Create view on SYSIBM.SYSROUTINES
   7 - Display views on SYSIBM.SYSROUTINES

   Catalog table/view for option 1:
     Owner  ===> SYSIBM           (default is SYSIBM)
     Name   ===> SYSROUTINES      (default is SYSROUTINES)

 User defined functions can also be managed from option 1.F
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 581. ***Manage Functions (ADB2ZF)*** *panel*

3. Specify option 1, and press Enter.

   The **Functions (ADB21F)** panel is displayed, as shown in the following figure.



```
DB2 Admin -------------------- DB2X Functions -------------------- Row 1 of 44
Command ===>                                                  Scroll ===> PAGE

Line commands:
 AH - Schema auth  A - Auth  DROP - Drop  AL - Alter   K - Package  PA - Parms
 RT - Return type  DIS - Display  STO - Stop  STA - Start  GEN - Generate DDL
 COM - Comment  I - Interpret  CRE - Create func  GR - Grant  JAR - JAVA detail
 DDL - Object DDL
                                                     D         S
                                            F        E E C P   Q S P E External
S    Schema   Name     Specific Name      O T Parms  T A F S F L R T S Name
     *        *        *                  * * *      * * * * * * * * * *
---- -------> -------> -----------------> - - ------ - - - - - - - - - --------
     ISTJE    +        SQL990208100338896 U S      2     N
     ISTJE    -        KR_MINUS           U S      2     N
     ISTJE    BLOB     SQL99020816075424# S S      1     Y
     ISTJE    CHAR     SQL990208160600039 S S      1     Y
     ISTJE    CLOB     SQL99020816074873# S S      1     Y
     ISTJE    D        SQL99020817171170M S S      1     Y
     ISTJE    DATE     SQL99020816083184# S S      1     Y
     ISTJE    DECIMAL  SQL99011815223541B S S      1     Y
     ISTJE    DECIMAL  SQL99021816281595J S S      1     Y
     ISTJE    DECIMAL  SQL99020817171173M S S      1     Y
```

Figure 582. ***Functions (ADB21F)*** *panel*

This panel displays information about all the user-defined functions in your Db2 subsystem.

The following columns are available on this panel:

S
: Input field where you enter one of the line commands listed on the panel.

Schema
: Schema of the function.

Name
Name of the function.

Specific Name
Specific name of the function.

O
Origin of the function, which is one of the following:

E
External

U
Sourced

S
System generated

Q
SQL

FT
Function type, which is one of the following:

C
Column

S
Scaler

T
Table

Parms
Number of parameters for the function.

DET
Whether the external function returns the same result when called using the same parameters. This field contains one of the following:

Y
Yes

N
No

blank
The routine is a function, but not an external function.

EA
Whether the external function changes the state of an object that Db2 does not manage. This field contains one of the following:

Y
Yes

N
No

blank
The routine is not an external function.

CF
Cast function, which is one of the following:

Y
Yes

N
No

PS
Parameter style, which is one of the following:

D
DB2SQL

G
General

N
General with nulls

J
Java

blank
Not external or user-defined function.

F
Fenced (applies if it is run separately from Db2).

SQL
Whether SQL statements are allowed, which is one of the following:

N
Contains no SQL statements

C
Contains SQL statements

R
Reads SQL data

M
Modifies SQL data

blank
Not applicable

SR
Whether the program should remain resident when it ends.

Y
Program remains resident

N
Program does not remain resident

blank
Not external or user-defined function.

PT
Program type, which is one of the following:

M
Main

S
Subroutine

blank
Not external or user-defined function.

ES
External security, which is one of the following:

D
Db2 address space user

U
User

C
Definer

blank
Not external or user-defined function.

External Name
Load module name for the stored procedure. This field is blank if it is not an external or user-defined function.

4. Optional: To alter functions, specify AL in the **Sel** column beside the function you want to alter, and press Enter.

Creating functions

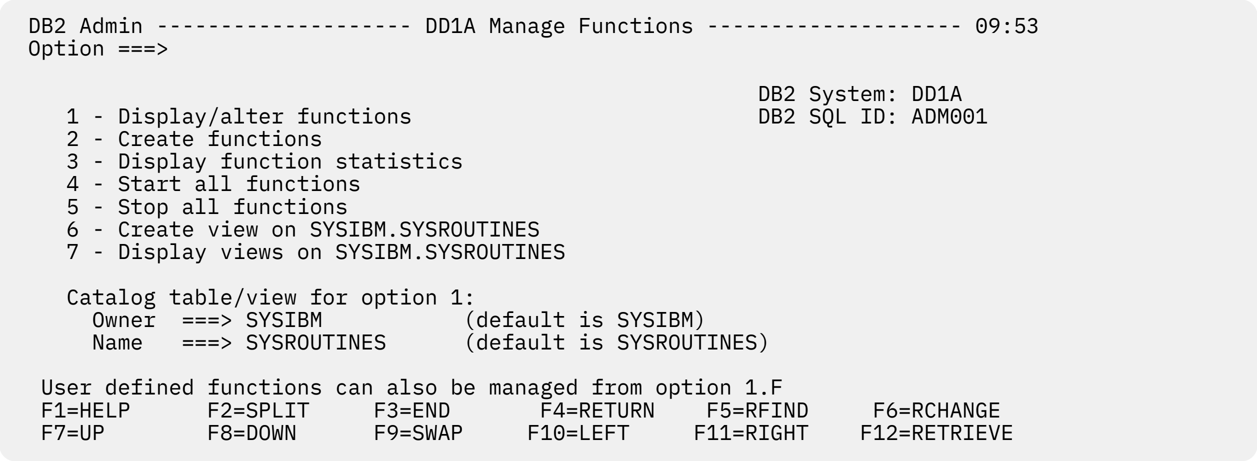
This procedure explains how to create new, user-defined functions.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option FM, and press Enter.

   The **Manage Functions (ADB2ZF)** panel is displayed, as shown in the following figure.

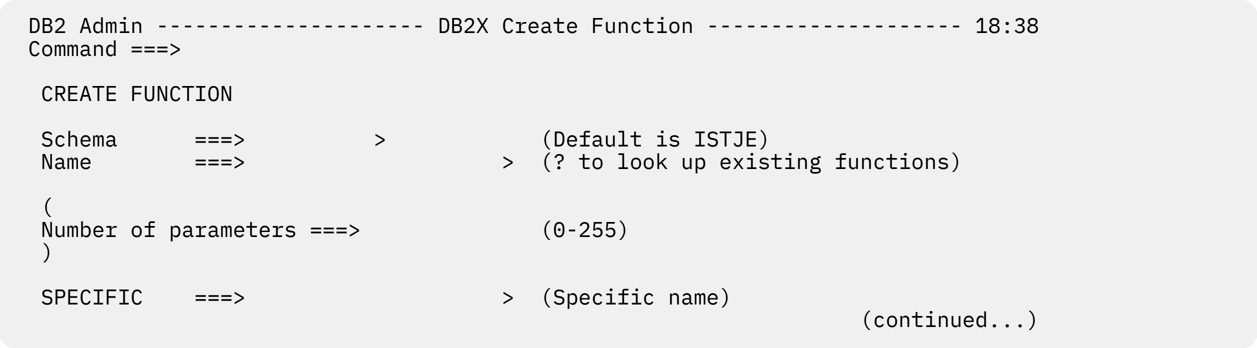


```
DB2 Admin -------------------- DD1A Manage Functions -------------------- 09:53
Option ===>

                                                         DB2 System: DD1A
   1 - Display/alter functions                           DB2 SQL ID: ADM001
   2 - Create functions
   3 - Display function statistics
   4 - Start all functions
   5 - Stop all functions
   6 - Create view on SYSIBM.SYSROUTINES
   7 - Display views on SYSIBM.SYSROUTINES

   Catalog table/view for option 1:
     Owner  ===> SYSIBM           (default is SYSIBM)
     Name   ===> SYSROUTINES      (default is SYSROUTINES)

 User defined functions can also be managed from option 1.F
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

*Figure 583. **Manage Functions (ADB2ZF)** panel*

3. Specify option 2, and press Enter.

   The **Create Function (ADB26CF)** panel is displayed, as shown in the following figure.



```
DB2 Admin --------------------- DB2X Create Function -------------------- 18:38
Command ===>

 CREATE FUNCTION

 Schema      ===>          >            (Default is ISTJE)
 Name        ===>                    >  (? to look up existing functions)

 (
 Number of parameters ===>              (0-255)
 )

 SPECIFIC    ===>                    >  (Specific name)
                                                                 (continued...)
```

*Figure 584. **Create Function (ADB26CF)** panel*

4. Enter the required parameters and press Enter to continue with the create operation, or press End to avoid creating a function.

   Db2 Admin Tool issues the SQL CREATE FUNCTION statement with the parameters you specify.

To create a new SQL scalar function:

Restriction: When creating SQL scalar functions, the maximum length of the return statement is 2MB (32,767KB).

a) Write the SQL scalar function as part of the CREATE statement.

b) Pre-compile, compile, and link the program.

c) If the program has SQL statements, bind a package.

d) Create the function to register it to Db2 and grant execute to authorize all appropriate users.

e) Use the function in application programs.

Displaying function statistics

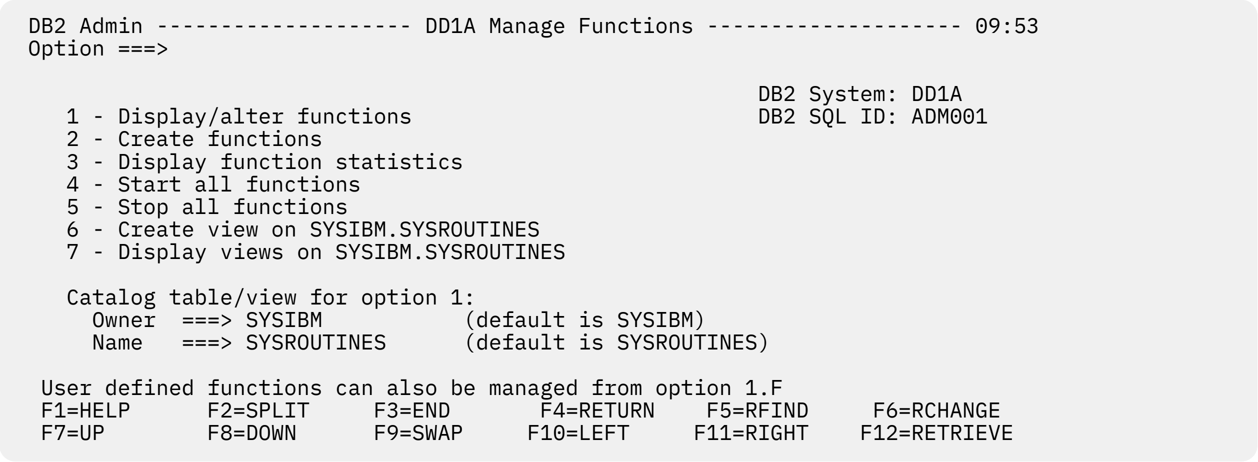
This procedure explains how to display function statistics.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option FM, and press Enter.

   The **Manage Functions (ADB2ZF)** panel is displayed, as shown in the following figure.

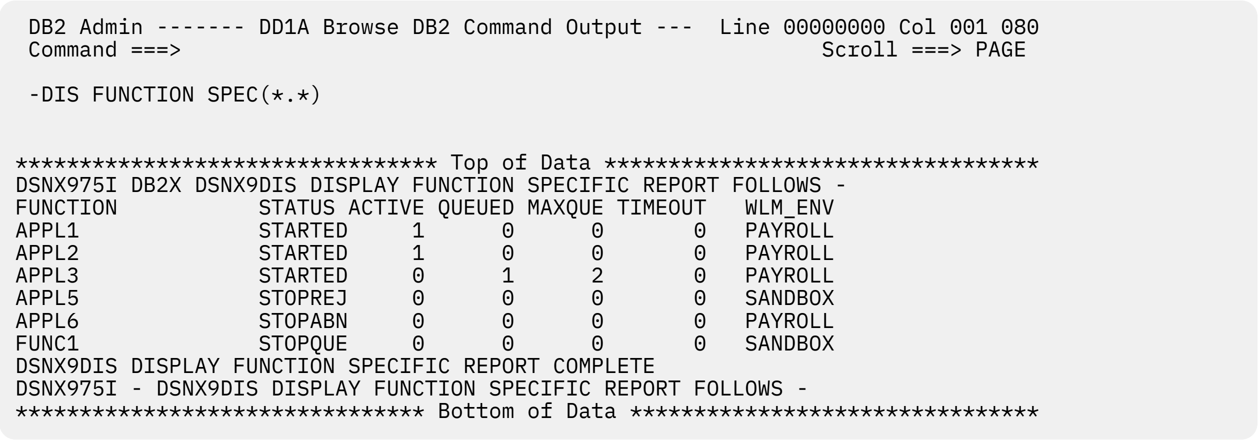


```
DB2 Admin -------------------- DD1A Manage Functions -------------------- 09:53
Option ===>

                                                         DB2 System: DD1A
   1 - Display/alter functions                           DB2 SQL ID: ADM001
   2 - Create functions
   3 - Display function statistics
   4 - Start all functions
   5 - Stop all functions
   6 - Create view on SYSIBM.SYSROUTINES
   7 - Display views on SYSIBM.SYSROUTINES

   Catalog table/view for option 1:
     Owner  ===> SYSIBM           (default is SYSIBM)
     Name   ===> SYSROUTINES      (default is SYSROUTINES)

 User defined functions can also be managed from option 1.F
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

*Figure 585. **Manage Functions (ADB2ZF)** panel*

3. Specify option 3, and press Enter.

   The **Display Function Statistics (ADB2DB2O)** panel, as shown in the following figure, is displayed. This panel displays statistics about external user-defined functions accessed by Db2 applications.



```
 DB2 Admin ------- DD1A Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

 -DIS FUNCTION SPEC(*.*)


********************************* Top of Data **********************************
DSNX975I DB2X DSNX9DIS DISPLAY FUNCTION SPECIFIC REPORT FOLLOWS -
FUNCTION           STATUS ACTIVE QUEUED MAXQUE TIMEOUT   WLM_ENV
APPL1              STARTED     1      0      0       0   PAYROLL
APPL2              STARTED     1      0      0       0   PAYROLL
APPL3              STARTED     0      1      2       0   PAYROLL
APPL5              STOPREJ     0      0      0       0   SANDBOX
APPL6              STOPABN     0      0      0       0   PAYROLL
FUNC1              STOPQUE     0      0      0       0   SANDBOX
DSNX9DIS DISPLAY FUNCTION SPECIFIC REPORT COMPLETE
DSNX975I - DSNX9DIS DISPLAY FUNCTION SPECIFIC REPORT FOLLOWS -
******************************** Bottom of Data ********************************
```

*Figure 586. **Display Function Statistics (ADB2DB2O)** panel*

When you press Enter, Db2 Admin Tool issues the -DIS FUNCTION SPEC(*.*) command.

Starting all functions

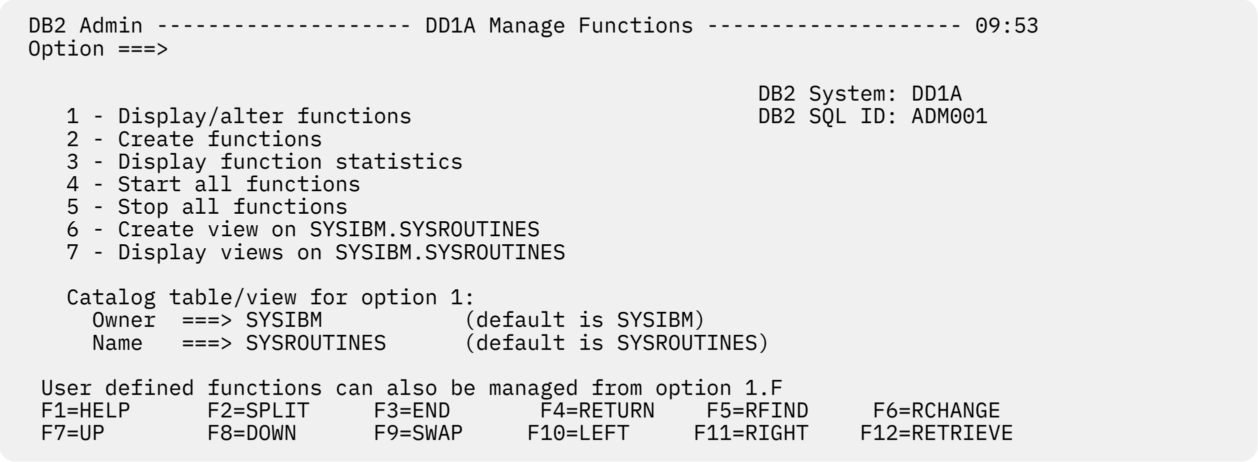
This procedure explains how to start all functions.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option FM, and press Enter.

   The **Manage Functions (ADB2ZF)** panel is displayed, as shown in the following figure.

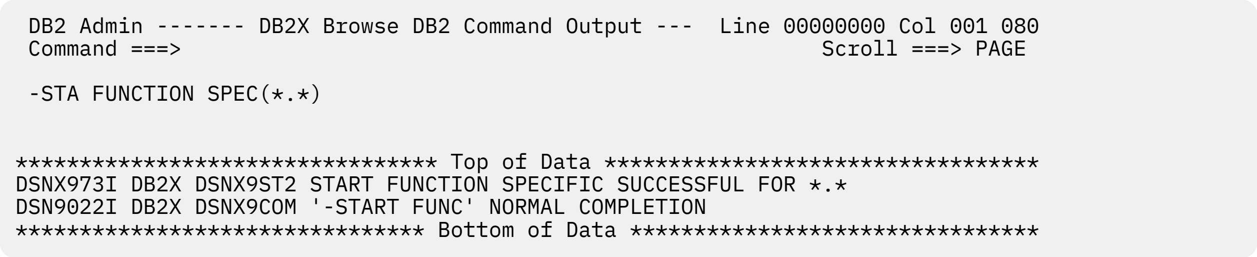


```
 DB2 Admin -------------------- DD1A Manage Functions -------------------- 09:53
 Option ===>

                                                         DB2 System: DD1A
    1 - Display/alter functions                          DB2 SQL ID: ADM001
    2 - Create functions
    3 - Display function statistics
    4 - Start all functions
    5 - Stop all functions
    6 - Create view on SYSIBM.SYSROUTINES
    7 - Display views on SYSIBM.SYSROUTINES

    Catalog table/view for option 1:
      Owner  ===> SYSIBM           (default is SYSIBM)
      Name   ===> SYSROUTINES      (default is SYSROUTINES)

  User defined functions can also be managed from option 1.F
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

*Figure 587. **Manage Functions (ADB2ZF)** panel*

3. Specify option 4, and press Enter.

   Db2 Admin Tool issues the -STA FUNCTION SPEC(*.*) command, and displays the status of the command in an ISPF edit session, as shown in the following figure.



```
 DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

 -STA FUNCTION SPEC(*.*)


********************************* Top of Data **********************************
DSNX973I DB2X DSNX9ST2 START FUNCTION SPECIFIC SUCCESSFUL FOR *.*
DSN9022I DB2X DSNX9COM '-START FUNC' NORMAL COMPLETION
******************************** Bottom of Data ********************************
```

Figure 588. Start All Functions panel (ADB2DB2O)

Stopping all functions

This procedure explains how to stop all functions.

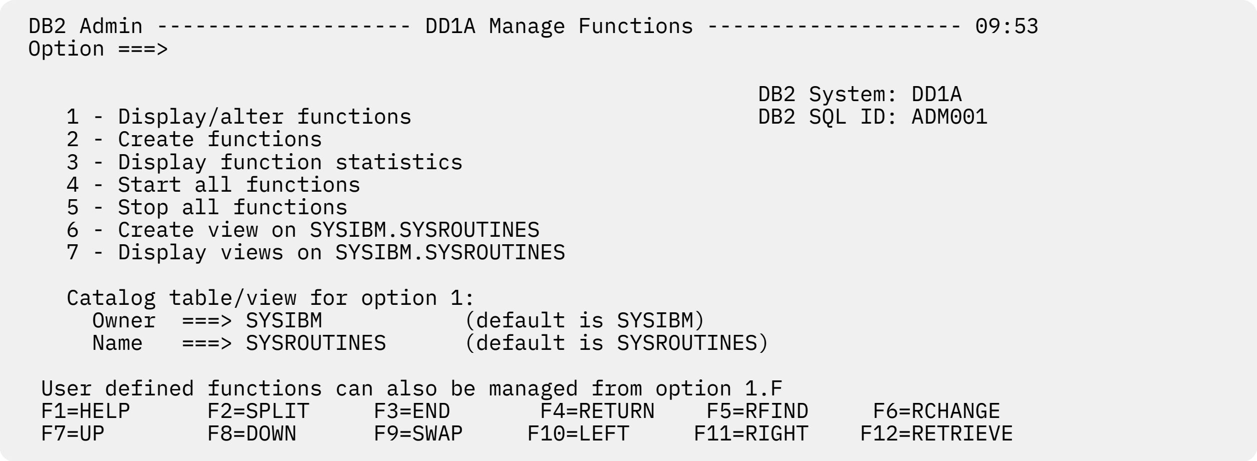
Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.

2. Specify option FM, and press Enter.

   The **Manage Functions (ADB2ZF)** panel is displayed, as shown in the following figure.

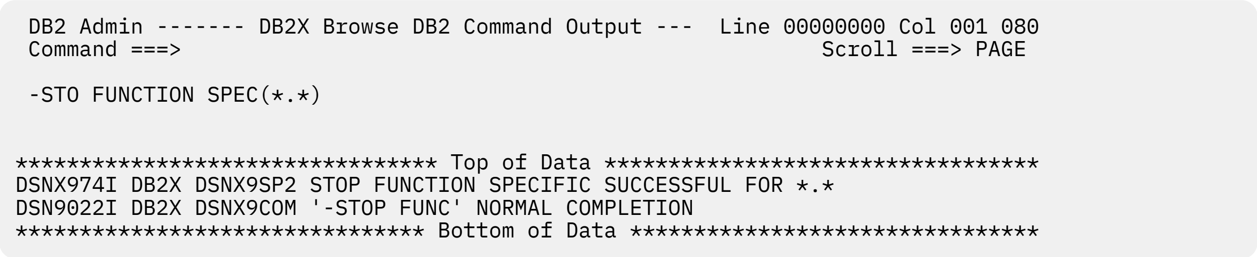


```
 DB2 Admin -------------------- DD1A Manage Functions -------------------- 09:53
 Option ===>

                                                          DB2 System: DD1A
    1 - Display/alter functions                           DB2 SQL ID: ADM001
    2 - Create functions
    3 - Display function statistics
    4 - Start all functions
    5 - Stop all functions
    6 - Create view on SYSIBM.SYSROUTINES
    7 - Display views on SYSIBM.SYSROUTINES

    Catalog table/view for option 1:
      Owner  ===> SYSIBM           (default is SYSIBM)
      Name   ===> SYSROUTINES      (default is SYSROUTINES)

  User defined functions can also be managed from option 1.F
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 589. Manage Functions panel (ADB2ZF)

3. Specify option 5, and press Enter.

   Db2 Admin Tool issues the -STO FUNCTION SPEC(*.*) command and displays the status of the command in an ISPF edit session, as shown in the following figure.



```
 DB2 Admin ------- DB2X Browse DB2 Command Output ---  Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE

 -STO FUNCTION SPEC(*.*)


********************************* Top of Data **********************************
DSNX974I DB2X DSNX9SP2 STOP FUNCTION SPECIFIC SUCCESSFUL FOR *.*
DSN9022I DB2X DSNX9COM '-STOP FUNC' NORMAL COMPLETION
******************************** Bottom of Data ********************************
```

Figure 590. Stop All Functions panel (ADB2DB2O)

Creating views of functions

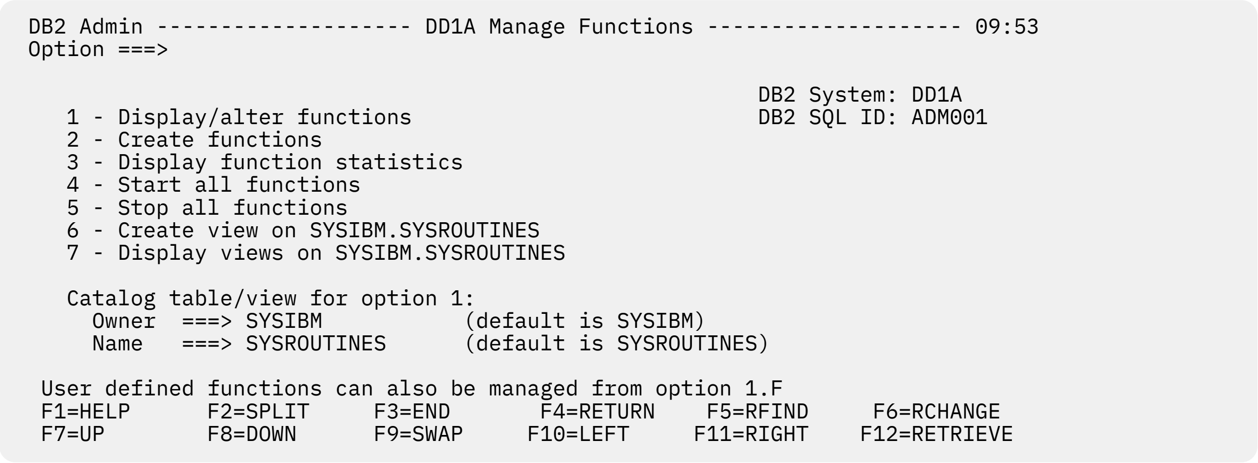
This procedure explains how to create a view of a function on SYSIBM.SYSROUTINES to enable other people to administer their own functions.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option FM, and press Enter.

   The **Manage Functions (ADB2ZF)** panel is displayed, as shown in the following figure.

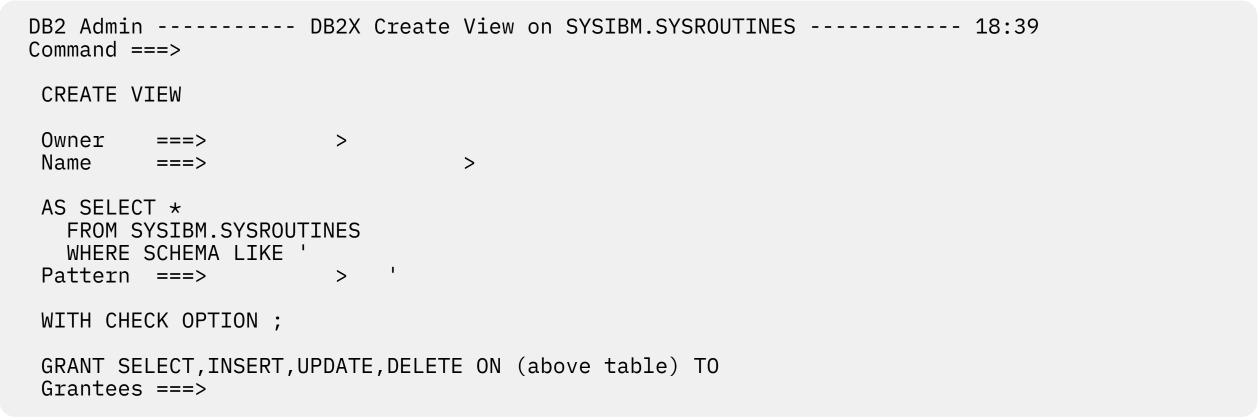


```
DB2 Admin -------------------- DD1A Manage Functions -------------------- 09:53
Option ===>

                                                         DB2 System: DD1A
   1 - Display/alter functions                           DB2 SQL ID: ADM001
   2 - Create functions
   3 - Display function statistics
   4 - Start all functions
   5 - Stop all functions
   6 - Create view on SYSIBM.SYSROUTINES
   7 - Display views on SYSIBM.SYSROUTINES

   Catalog table/view for option 1:
     Owner  ===> SYSIBM           (default is SYSIBM)
     Name   ===> SYSROUTINES      (default is SYSROUTINES)

 User defined functions can also be managed from option 1.F
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

*Figure 591. **Manage Functions (ADB2ZF)** panel*

3. Specify option 6, and press Enter.

   The **Create View on SYSIBM.SYSROUTINES (ADB2ZF6)** panel is displayed, as shown in the following figure. This panel enables you to define a view for all procedures with the (LIKE) pattern you define.



```
DB2 Admin ----------- DB2X Create View on SYSIBM.SYSROUTINES ------------ 18:39
Command ===>

 CREATE VIEW

 Owner    ===>          >
 Name     ===>                    >

 AS SELECT *
   FROM SYSIBM.SYSROUTINES
   WHERE SCHEMA LIKE '
 Pattern  ===>          >   '

 WITH CHECK OPTION ;

 GRANT SELECT,INSERT,UPDATE,DELETE ON (above table) TO
 Grantees ===>
```

*Figure 592. **Create View on SYSIBM.SYSROUTINES (ADB2ZF6)** panel*

4. Fill in the fields on this panel to create a view, for example, Define view ABC.FUNCTIONS as a view on SYSIBM.SYSROUTINES WHERE SCHEMA LIKE 'ABC.

   View ABC.FUNCTIONS contain all user-defined functions in schemas starting with ABC. In addition, you can issue GRANT SELECT, INSERT, UPDATE, or DELETE statements on the view to a list of authorization IDs (grantees).

Displaying views of functions

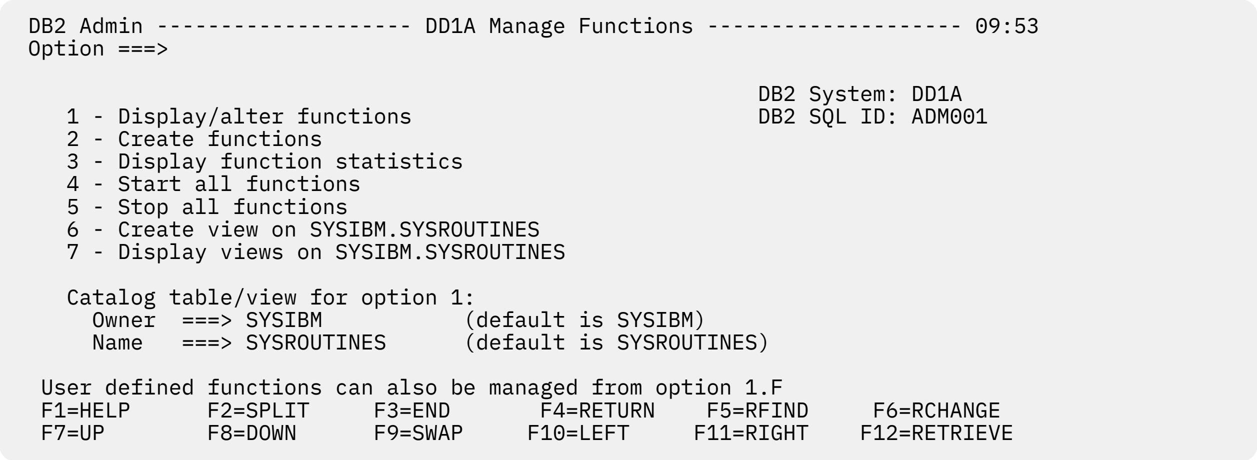
This procedure explains how to display views of functions on SYSIBM.SYSROUTINES.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option FM, and press Enter.

   The **Manage Functions** panel is displayed, as shown in the following figure.

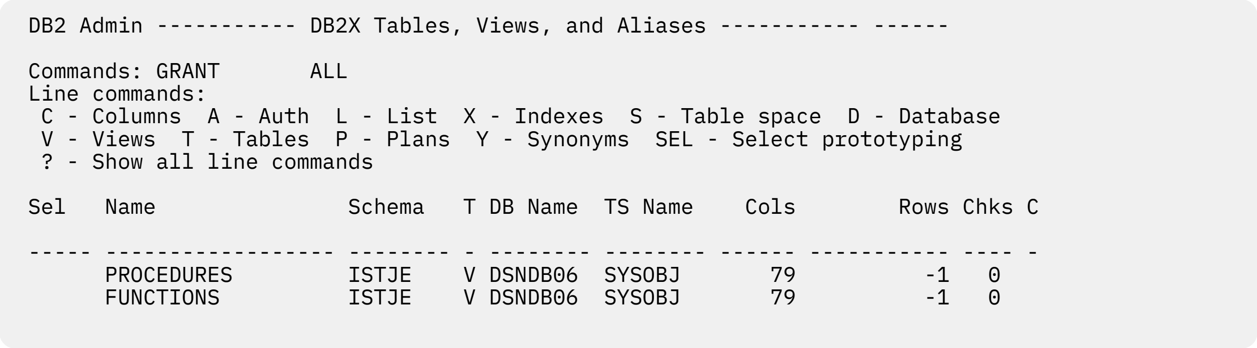


```
DB2 Admin -------------------- DD1A Manage Functions -------------------- 09:53
Option ===>

                                                         DB2 System: DD1A
   1 - Display/alter functions                           DB2 SQL ID: ADM001
   2 - Create functions
   3 - Display function statistics
   4 - Start all functions
   5 - Stop all functions
   6 - Create view on SYSIBM.SYSROUTINES
   7 - Display views on SYSIBM.SYSROUTINES

   Catalog table/view for option 1:
     Owner  ===> SYSIBM           (default is SYSIBM)
     Name   ===> SYSROUTINES      (default is SYSROUTINES)

 User defined functions can also be managed from option 1.F
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCHANGE
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 593. Manage Functions panel (ADB2ZF)

3. Specify option 7, and press Enter.

   The **Tables, Views, and Aliases (ADB21T)** panel is displayed, as shown in the following figure. This panel displays the views that exist on SYSIBM.SYSROUTINES.



```
DB2 Admin ----------- DB2X Tables, Views, and Aliases ----------- ------

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

Sel   Name              Schema   T DB Name  TS Name    Cols        Rows Chks C

----- ------------------ -------- - -------- -------- ------ ----------- ---- -
      PROCEDURES        ISTJE    V DSNDB06  SYSOBJ       79          -1   0
      FUNCTIONS         ISTJE    V DSNDB06  SYSOBJ       79          -1   0
```

Figure 594. Tables, Views, and Aliases panel showing views on SYSIBM.SYSROUTINES (ADB21T)

The panel being displayed is the same panel you get if you use option 1.T and option Z.PM.7.

Db2 subsystem backup and recovery

The Db2 subsystem can be backed up, and jobs can be set up to specify a particular point in time to which to recover the subsystem or to recover the Db2 subsystem to a point in time.

Deprecation notice: System-level backups are deprecated in Db2 Admin Tool.For more information, see "Deprecated functions and functions that are no longer supported in Db2 Admin Tool 12.1" on page 64.

Subsystem-level backups and recovery are possible only with the Db2 for z/OS BACKUP SYSTEM and RESTORE SYSTEM utilities. Both utilities invoke z/OS DFSMShsm (Version 1 Release 5 or above). The BACKUP SYSTEM utility uses copy pools, which are new constructs in z/OS DFSMShsm. The RESTORE

SYSTEM utility uses data that is copied by the BACKUP SYSTEM utility, and the data sets that are to be recovered must be SMS-managed data sets.

You can submit the batch job that Db2 Admin Tool creates for backing up the system directly from Db2 Admin Tool. You cannot directly submit the other batch jobs that Db2 Admin Tool creates for specifying a particular time to which to recover the subsystem or for recovering the subsystem. These batch jobs cannot be run from Db2 Admin Tool.

Backing up the Db2 subsystem

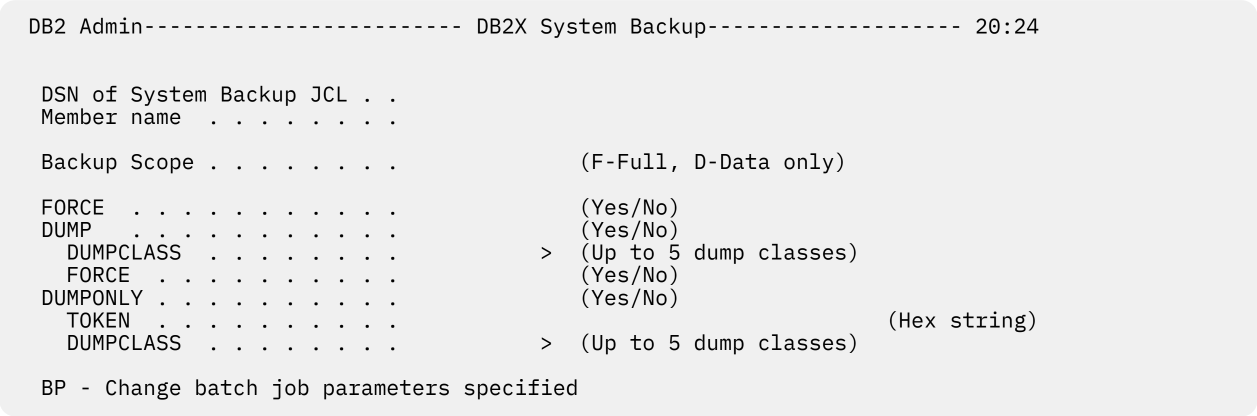
This procedure explains how to back up the Db2 subsystem.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option SB, and press Enter.

   The **System Backup (ADB2ZSB)** panel is displayed, as shown in the following figure.



```
DB2 Admin------------------------- DB2X System Backup-------------------- 20:24


 DSN of System Backup JCL . .
 Member name  . . . . . . . .

 Backup Scope . . . . . . . .              (F-Full, D-Data only)

 FORCE  . . . . . . . . . . .              (Yes/No)
 DUMP   . . . . . . . . . . .              (Yes/No)
   DUMPCLASS  . . . . . . . .           >  (Up to 5 dump classes)
   FORCE  . . . . . . . . . .              (Yes/No)
 DUMPONLY . . . . . . . . . .              (Yes/No)
   TOKEN  . . . . . . . . . .                                        (Hex string)
   DUMPCLASS  . . . . . . . .           >  (Up to 5 dump classes)

 BP - Change batch job parameters specified
```

*Figure 595. **System Backup (ADB2ZSB)** panel*

3. Specify values for the **DSN of System Backup JCL** and **Member name** fields to identify where the generated JCL is to be stored.
4. Specify values for the fields related to other copy options and backup scope.

   Depending on the level of Db2 that you are using, some of the fields on this panel might be hidden. See the online help for a description of the fields that are displayed.
5. Press Enter.

   Db2 Admin Tool displays the generated JCL for the backup job.
6. To backup the system, submit the JCL.

Specifying a point in time to which to recover

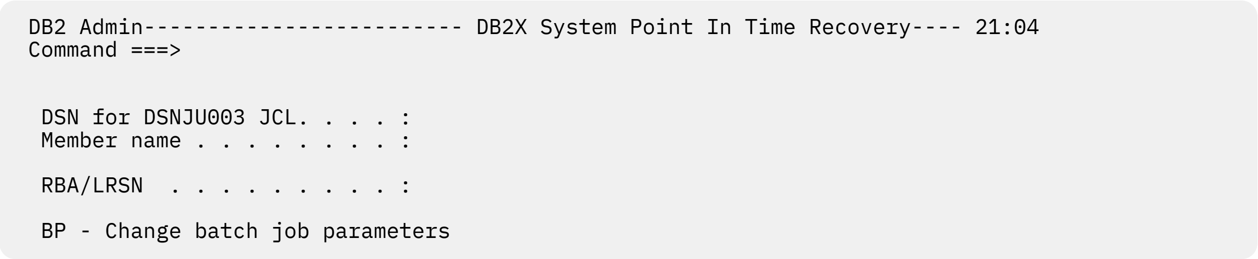
You can set up a batch job that will specify a particular time to which to recover the Db2 system.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option PT, and press Enter.

   The **System Point in Time Recovery** panel is displayed, as shown in the following figure.



```
DB2 Admin------------------------ DB2X System Point In Time Recovery---- 21:04
Command ===>


 DSN for DSNJU003 JCL. . . . :
 Member name . . . . . . . . :

 RBA/LRSN  . . . . . . . . . :

 BP - Change batch job parameters
```

*Figure 596. **System Point in Time Recovery** panel*

3. Specify values for the following fields:

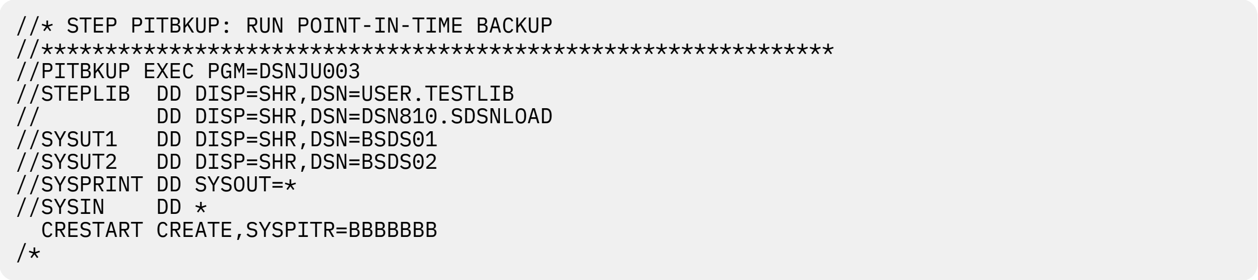
   DSN
   Member name
   The name of the data set and member in which the generated JCL is to be stored.

   RBA/LRSN

   - To set a point in time for recovery of a non-data sharing member, specify an RBA value.
   - To set a point in time for recovery of a data sharing member, specify an LRSN value.

4. Press Enter.

   Db2 Admin Tool displays the generated JCL for the job, as shown in the following figure.



```
//* STEP PITBKUP: RUN POINT-IN-TIME BACKUP
//**************************************************************
//PITBKUP EXEC PGM=DSNJU003
//STEPLIB  DD DISP=SHR,DSN=USER.TESTLIB
//         DD DISP=SHR,DSN=DSN810.SDSNLOAD
//SYSUT1   DD DISP=SHR,DSN=BSDS01
//SYSUT2   DD DISP=SHR,DSN=BSDS02
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  CRESTART CREATE,SYSPITR=BBBBBBB
/*
```

Figure 597. Example of Generated JCL for System Point in Time Recovery

5. Edit the generated JCL to specify the appropriate BSDS data set names in //SYSUT1 and //SYSUT2.
6. Save the JCL for the batch job.

   The batch job cannot be submitted directly after it is created. It cannot be run from Db2 Admin Tool.

Recovering the Db2 subsystem

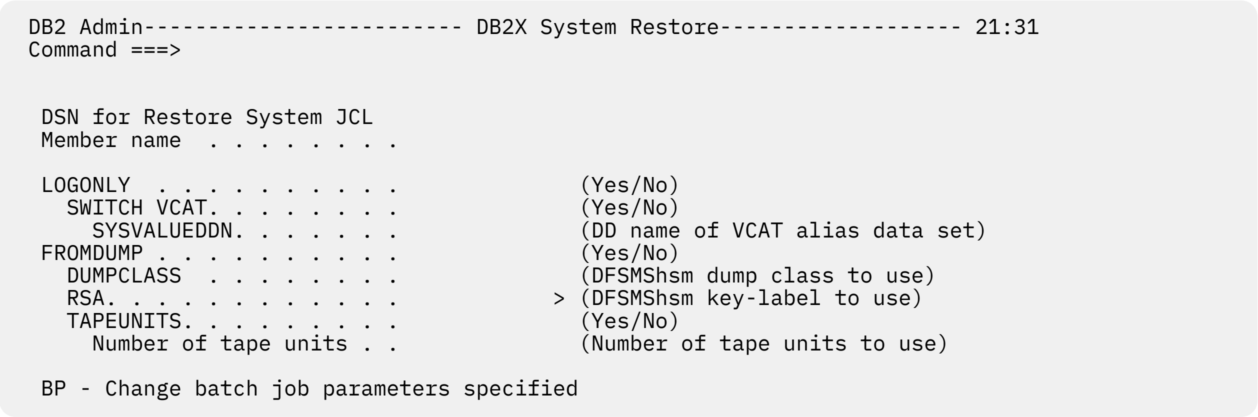
You can set up a batch job that will recover the Db2 subsystem to a previous point in time.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option SR, and press Enter.

   The **System Restore (ADB2ZSR)** panel is displayed, as shown in the following figure.



```
DB2 Admin------------------------- DB2X System Restore------------------ 21:31
Command ===>


 DSN for Restore System JCL
 Member name  . . . . . . . .

 LOGONLY  . . . . . . . . . .             (Yes/No)
   SWITCH VCAT. . . . . . . .             (Yes/No)
     SYSVALUEDDN. . . . . . .             (DD name of VCAT alias data set)
 FROMDUMP . . . . . . . . . .             (Yes/No)
   DUMPCLASS  . . . . . . . .             (DFSMShsm dump class to use)
   RSA. . . . . . . . . . . .           > (DFSMShsm key-label to use)
   TAPEUNITS. . . . . . . . .             (Yes/No)
     Number of tape units . .             (Number of tape units to use)

 BP - Change batch job parameters specified
```

*Figure 598. **System Restore (ADB2ZSR)** panel*

3. Specify values for the **DSN for Restore System JCL** and **Member name** fields to identify where the generated JCL is to be stored, and specify values for any other appropriate options.

   Depending on the level of Db2 that you are using, some of the fields on this panel might be hidden. See the online help for a description of the fields that are displayed.
4. Press Enter.

   Db2 Admin Tool displays the generated JCL for the job, which invokes the RESTORE SYSTEM utility.
5. Save the JCL for the batch job.

   Restriction: The batch job cannot be submitted directly after it is created. It cannot be run from Db2 Admin Tool.

Stopping Db2

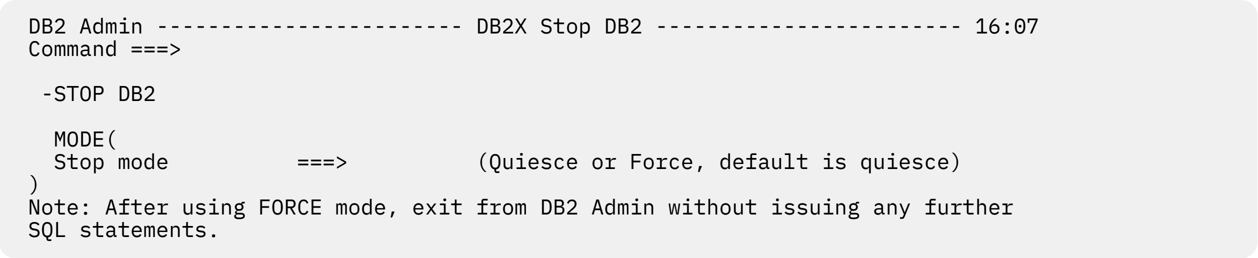
This procedure explains how to stop the Db2 subsystem.

Procedure

1. On the **DB2 Administration Menu (ADB2)** panel, specify Z, and press Enter.

   The **System Administration (ADB2Z)** panel is displayed.
2. Specify option 2S, and press Enter.

   The **Stop DB2 (ADB2Z2S)** panel is displayed, as shown in the following figure.



```
DB2 Admin ------------------------ DB2X Stop DB2 ------------------------ 16:07
Command ===>

 -STOP DB2

  MODE(
  Stop mode          ===>          (Quiesce or Force, default is quiesce)
)
Note: After using FORCE mode, exit from DB2 Admin without issuing any further
SQL statements.
```

*Figure 599. **Stop DB2 (ADB2Z2S)** panel*

3. In the **Stop mode** field, specify `Quiesce` or `Force`, and press Enter.

   Db2 Admin Tool accomplishes this task by issuing the Db2 command STOP DB2.

   The information that Db2 Admin Tool returns to you from the command is in ISPF browse format.

Chapter 6. Advanced topics

The information in this section is for experienced Db2 Admin Tool users. It explains how to write and modify applications, use copies of the Db2 catalog, and use distributed support.

Defining your own primary commands

When you define a primary command in Db2 Admin Tool, it can then be issued from any command line in the product.

About this task

The default primary command list that ships with the product is stored in an ISPF table in *hlq*.SADBTLIB(ADBSCMDL). You cannot edit this list. However, you can use the panel interface in Db2 Admin Tool to define your own primary commands in one or more new lists. This process is described in the following procedure.

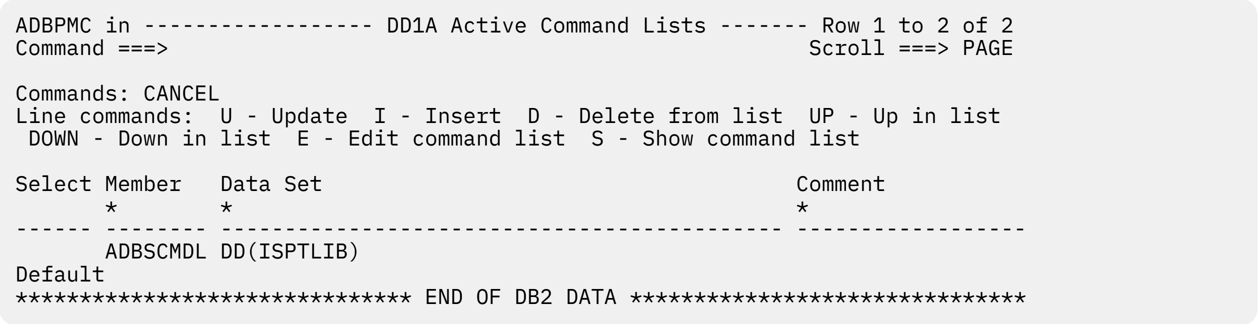
You can also define a list of commands that can be used by all users by specifying the installation command list in TCz. On the **Product Parameters (CCQPPRD)** panel, specify the user-defined command library in the **User CMDS dataset(mbr)** field. The specified data set must have RECFM=FB, and the member must be an ISPF table that conforms to the format of ADBSCMDL. This list is also not editable.

Procedure

To define your own primary commands:

1. From any panel in Db2 Admin Tool, issue the CMDS primary command, and press Enter.

   The **Active Command Lists (ADBPMC)** panel is displayed:

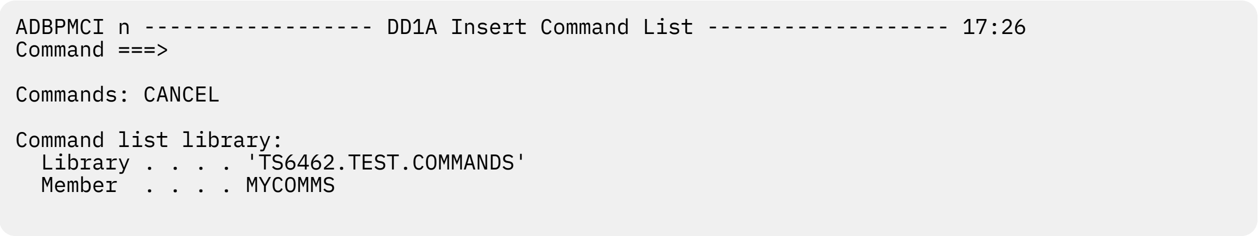


```
ADBPMC in ------------------ DD1A Active Command Lists ------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

Commands: CANCEL
Line commands:  U - Update  I - Insert  D - Delete from list  UP - Up in list
 DOWN - Down in list  E - Edit command list  S - Show command list

Select Member   Data Set                                     Comment
       *        *                                            *
------ -------- -------------------------------------------- ------------------
       ADBSCMDL DD(ISPTLIB)
Default
******************************* END OF DB2 DATA *******************************
```

   *Figure 600. **Active Command Lists (ADBPMC)** panel*

   The row that has `Default` in the **Comment** column is the default command list that shipped with the product. If a row has `Installation def.` in the **Comment** column, that list is the installation command list that was added during customization of the product. Neither of these lists can be edited.

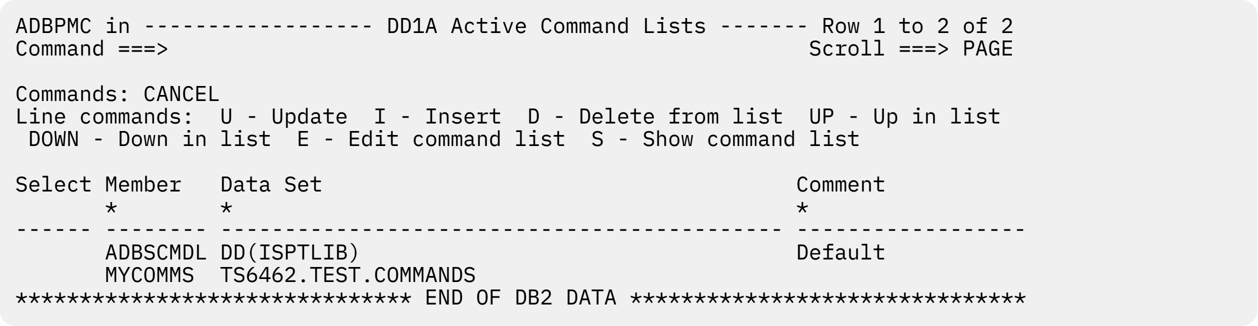
2. Specify the I line command to insert a new list, and press Enter.
3. On the **Insert Command List (ADBPMCI)** panel, specify the library and member name for your new list, and press Enter. The specified data set must already exist.



```
ADBPMCI n ------------------ DD1A Insert Command List ------------------- 17:26
Command ===>

Commands: CANCEL

Command list library:
  Library . . . . 'TS6462.TEST.COMMANDS'
  Member  . . . . MYCOMMS
```

Figure 601. ***Insert Command List (ADBPMCI)*** *panel*

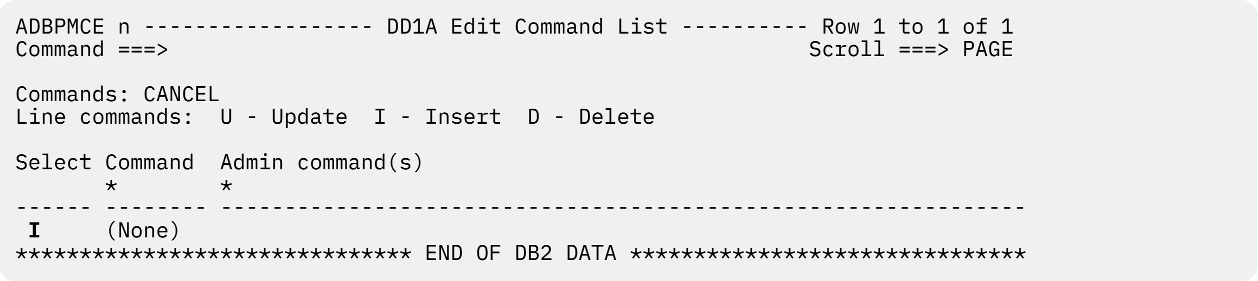
When you press Enter, your new list is displayed on the **Active Command Lists (ADBPMC)** panel:



```
ADBPMC in ------------------ DD1A Active Command Lists ------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

Commands: CANCEL
Line commands:  U - Update  I - Insert  D - Delete from list  UP - Up in list
 DOWN - Down in list  E - Edit command list  S - Show command list

Select Member   Data Set                                     Comment
       *        *                                            *
------ -------- -------------------------------------------- ------------------
       ADBSCMDL DD(ISPTLIB)                                  Default
       MYCOMMS  TS6462.TEST.COMMANDS
******************************* END OF DB2 DATA *******************************
```

Figure 602. ***Active Command Lists (ADBPMC)*** *panel with a user-defined list*

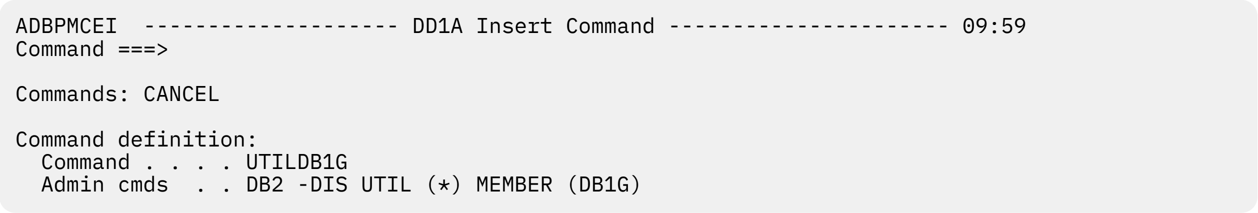
4. Specify the E line command next to your new list to edit it, and press Enter.
5. On the **Edit Command List (ADBPMCE)** panel, issue the I line command to insert a new command, and press Enter:



```
ADBPMCE n ------------------ DD1A Edit Command List ---------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Commands: CANCEL
Line commands:  U - Update  I - Insert  D - Delete

Select Command  Admin command(s)
       *        *
------ -------- ---------------------------------------------------------------
 I     (None)
******************************* END OF DB2 DATA *******************************
```

Figure 603. ***Edit Command List (ADBPMCE)*** *panel*

6. On the **Insert Command (ADBPMCEI)** panel, specify the name of your new command and the Db2 Admin Tool command that you want to execute when that new command is issued, and press Enter:

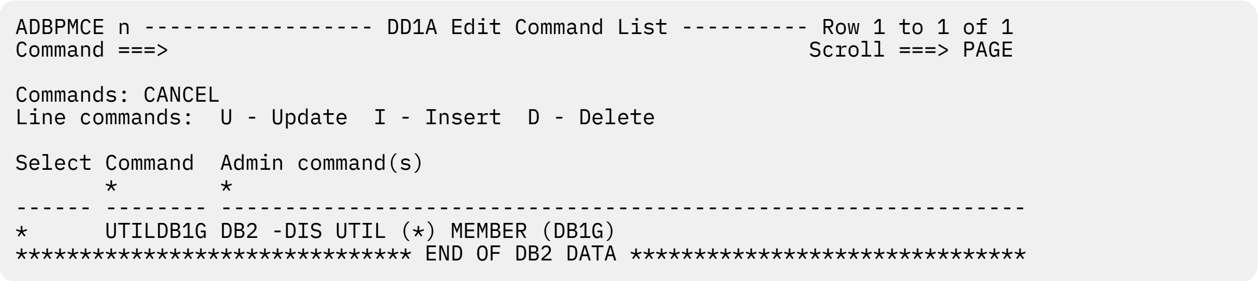


```
ADBPMCEI  -------------------- DD1A Insert Command --------------------- 09:59
Command ===>

Commands: CANCEL

Command definition:
  Command . . . . UTILDB1G
  Admin cmds  . . DB2 -DIS UTIL (*) MEMBER (DB1G)
```

Figure 604. ***Insert Command (ADBPMCEI)*** *panel*

- The command name (in the **Command** field) can be up to 8 characters.
- The command to be executed (in the **Admin cmds** field) can be any valid Db2 Admin Tool primary command, such as an SQL statement, Db2 command, or ISPF statement. For a complete list of valid commands, see the online help (PF1) for the main **DB2 Administration Menu (ADB2)** panel.

When you press Enter, your new command is listed on the **Edit Command List (ADBPMCE)** panel:



```
ADBPMCE n ------------------ DD1A Edit Command List ---------- Row 1 to 1 of 1
Command ===>                                                   Scroll ===> PAGE

Commands: CANCEL
Line commands:  U - Update  I - Insert  D - Delete

Select Command  Admin command(s)
       *        *
------ -------- ----------------------------------------------------------------
*      UTILDB1G DB2 -DIS UTIL (*) MEMBER (DB1G)
******************************* END OF DB2 DATA *******************************
```

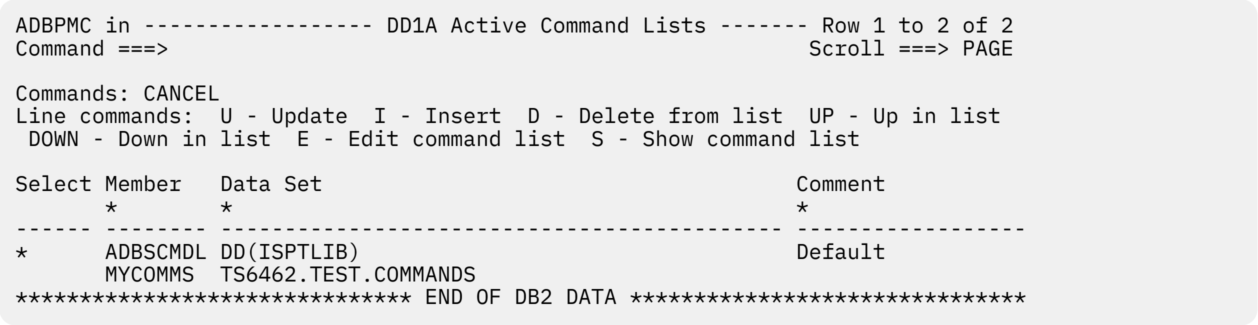
Figure 605. ***Edit Command List (ADBPMCE)*** *panel with a command defined*

7. Repeat the previous step as needed to add all of your commands.

   You can also create additional lists and add commands to those lists as needed.

8. Exit (PF3) back to the **Active Command Lists (ADBPMC)** panel.

   Note that the order of rows on this panel determines which command list takes precedence. The higher the row, the higher the precedence. In the following example, ADBSCMDL takes precedence over MYCOMMS. If the same command is defined in both lists, the command defined in ADBSCMDL will be used.



```
ADBPMC in ------------------ DD1A Active Command Lists ------- Row 1 to 2 of 2
Command ===>                                                   Scroll ===> PAGE

Commands: CANCEL
Line commands:  U - Update  I - Insert  D - Delete from list  UP - Up in list
 DOWN - Down in list  E - Edit command list  S - Show command list

Select Member   Data Set                                     Comment
       *        *                                            *
------ -------- -------------------------------------------- -------------------
*      ADBSCMDL DD(ISPTLIB)                                  Default
       MYCOMMS  TS6462.TEST.COMMANDS
******************************* END OF DB2 DATA *******************************
```

Figure 606. ***Active Command Lists (ADBPMC)*** *panel with multiple command lists*

   To change the order of these lists (and therefore the precedence), use the UP and DOWN line commands.

9. Exit (PF3) this panel.

   Message ADB266I confirms that your changes are saved:

```
ADB266I User command list refreshed
```

Results

You can now issue any of the primary commands that you defined from any panel in Db2 Admin Tool.

Related information

Video: Defining your own primary commands

Viewing primary command lists

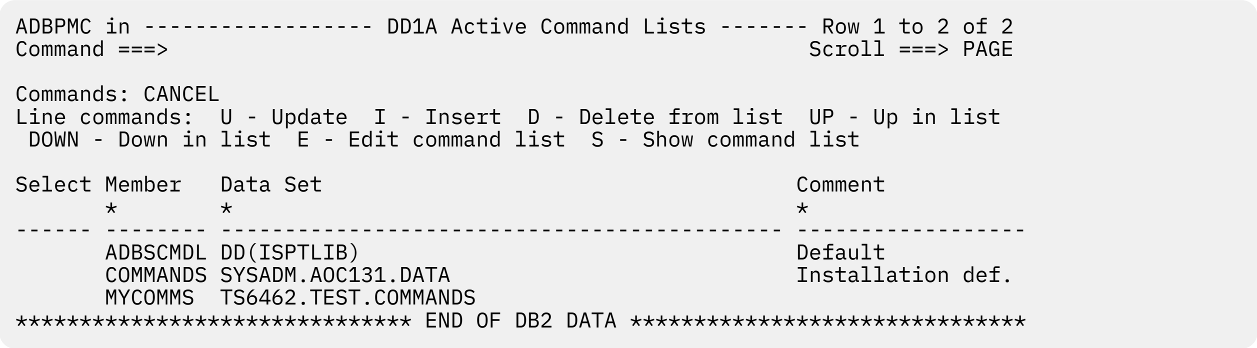
A default primary command list, ADBSCMDL, is provided by Db2 Admin Tool. Additionally, your installation might have an installation command list that was specified during customization. You can view both of these lists from the Db2 Admin Tool panel interface, but you cannot edit them. You can also view your own user-defined command lists from the panel interface; those lists can be edited.

Procedure

To view command lists:

1. From any panel in Db2 Admin Tool, issue the CMDS primary command, and press Enter.

The **Active Command Lists (ADBPMC)** panel lists any defined command lists, including the default list ADBSCMDL and the installation default.

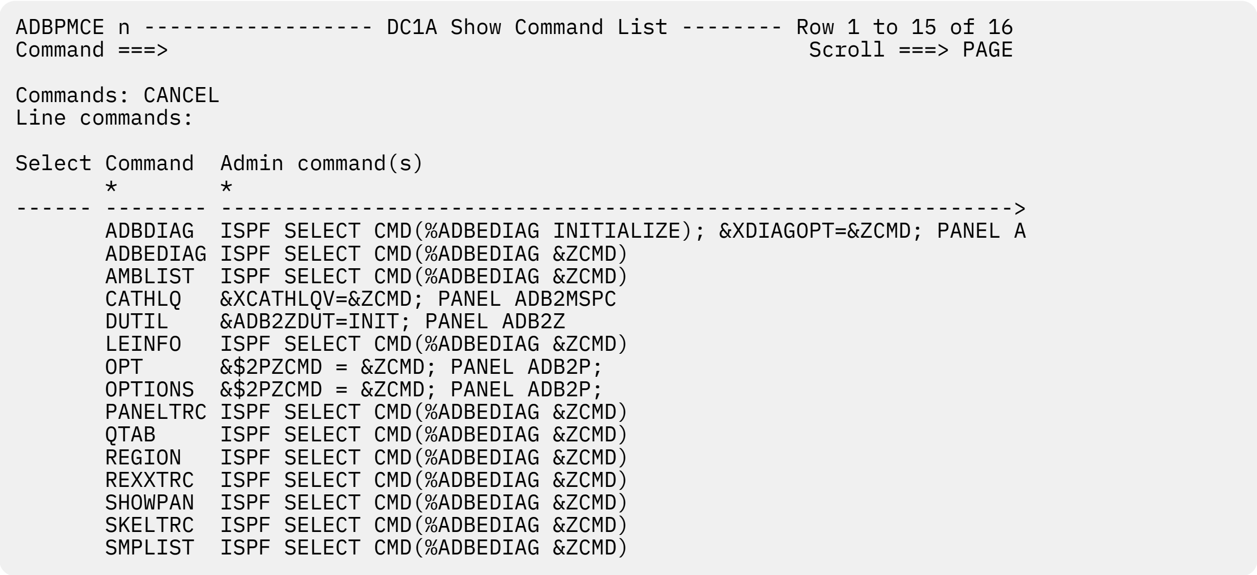


```
ADBPMC in ------------------ DD1A Active Command Lists ------- Row 1 to 2 of 2
Command ===>                                                  Scroll ===> PAGE

Commands: CANCEL
Line commands:  U - Update  I - Insert  D - Delete from list  UP - Up in list
 DOWN - Down in list  E - Edit command list  S - Show command list

Select Member   Data Set                                     Comment
       *        *                                            *
------ -------- -------------------------------------------- ------------------
       ADBSCMDL DD(ISPTLIB)                                  Default
       COMMANDS SYSADM.AOC131.DATA                           Installation def.
       MYCOMMS  TS6462.TEST.COMMANDS
******************************* END OF DB2 DATA *******************************
```

*Figure 607. **Active Command Lists (ADBPMC)** panel*

2. Issue the S line command next to the list that you want to view, and press Enter.

   The **Show Command List (ADBPMCE)** panel displays all of the commands in the selected list:



```
ADBPMCE n ------------------ DC1A Show Command List -------- Row 1 to 15 of 16
Command ===>                                                  Scroll ===> PAGE

Commands: CANCEL
Line commands:

Select Command  Admin command(s)
       *        *
------ -------- --------------------------------------------------------------->
       ADBDIAG  ISPF SELECT CMD(%ADBEDIAG INITIALIZE); &XDIAGOPT=&ZCMD; PANEL A
       ADBEDIAG ISPF SELECT CMD(%ADBEDIAG &ZCMD)
       AMBLIST  ISPF SELECT CMD(%ADBEDIAG &ZCMD)
       CATHLQ   &XCATHLQV=&ZCMD; PANEL ADB2MSPC
       DUTIL    &ADB2ZDUT=INIT; PANEL ADB2Z
       LEINFO   ISPF SELECT CMD(%ADBEDIAG &ZCMD)
       OPT      &$2PZCMD = &ZCMD; PANEL ADB2P;
       OPTIONS  &$2PZCMD = &ZCMD; PANEL ADB2P;
       PANELTRC ISPF SELECT CMD(%ADBEDIAG &ZCMD)
       QTAB     ISPF SELECT CMD(%ADBEDIAG &ZCMD)
       REGION   ISPF SELECT CMD(%ADBEDIAG &ZCMD)
       REXXTRC  ISPF SELECT CMD(%ADBEDIAG &ZCMD)
       SHOWPAN  ISPF SELECT CMD(%ADBEDIAG &ZCMD)
       SKELTRC  ISPF SELECT CMD(%ADBEDIAG &ZCMD)
       SMPLIST  ISPF SELECT CMD(%ADBEDIAG &ZCMD)
```

*Figure 608. **Show Command List (ADBPMCE)** panel*

   If the list is a user-defined command list and you want to edit it, exit back to the **Active Command Lists (ADBPMC)** panel and use the E line command to edit the list.

Defining your own line commands and primary commands for specific panels

You can define your own Db2 Admin Tool commands for each panel. For example, you might want to define line commands to start another ISPF-based tool with parameters from the current row, to display the contents of related tables, or to change the contents of the displayed row by using an SQL statement.

About this task

By using the ADBLCMDS command table, you can define line commands and primary commands for specific panels. The ADBLCMDS ISPF table has the following columns:

Table 47. ADBLCMDS columns

Column	Description
CPAN	The name of the panel for which the command is defined.
CMD	The command. The command must be the key in the table. If you set CMD to *MYCMD, MYCMD can be used as a primary command on the panel. The asterisk (*) indicates that the command is a primary command. The combination of the CPAN value and the CMD value must be unique for each row in the table.
DESCR	A description of the command. This description is displayed if you enter a question mark (?) to request further information.
SQL	The SQL statement that is run for this command. If PAN is also set, the result is displayed on that panel.
PAN	The panel to display as a result of this command.
ISPF	The ISPF statement that is run for this command. For example, specify `SELECT CMD(execname)` to run an exec.
ACMD	The Db2 Admin Tool command that is run for this command.

Important: If you have commands defined in ISPF line command tables that correspond to each panel, you must move those commands to the ADBLCMDS table.

Prior to the introduction of the ADBLCMDS table, commands were defined by running REXX execs to produce ISPF line command tables for each panel. That method is no longer supported. Instead, you must define those line commands in the ADBLCMDS table by using the following procedure. The ADBLCMDS table has the same columns as the ISPF line command tables plus one additional column to specify the panel to which the command applies.

Procedure

To define your own commands for a panel:

1. Copy the sample exec ADBLCMDS (in the SAMP library) and modify it to define your own line commands or primary commands for any panel.

   The ADBLCMDS exec generates the ISPF command table ADBLCMDS, which contains the user-defined commands.

   Use the examples and instructions in the exec to modify the exec as needed. For a description of the columns that you can define for each command, see Table 47 on page 1043.

   For example, to define primary command mylist on panel ADB21, set the ADBLCMDS columns as follows:

   CPAN=ADB21
   CMD=*MYLIST
   ISPF=SELECT CMD(%MYLIST &$TABLE)

   Note that ISPF=%MYLIST &$TABLE can also be used, because % is an alias for SELECT CMD(% ....). Also note that variable &$TABLE will contain the name of the currently displayed ISPF table.
2. Run the ADBLCMDS exec.

   Use the command `TSO EX '`*`dsnname`*`(`*`execname`*`)`, where *dsname* is data set name and *execname* is the member name to which you copied ADBLCMDS.
3. Copy the ADBLCMDS table to the ISPF table library that is used by the Db2 Admin Tool session. For example, DMTOOL.SADBPLIB

Results

When you issue the ? line command on any panel, Db2 Admin Tool displays both the line commands that are predefined for the panel and the line commands that you defined in ADBLCMDS for the panel.

Db2 Admin Tool application development

You can use Db2 Admin Tool to create your own applications and tools, and you can extend existing applications.

The tasks are the same for both creating and extending applications.

The application development process

Db2 Admin Tool allows you to add new line commands to existing panels and to develop new applications by using Db2 Admin Tool as the dialog driver and interface to DB2.

You can add new Db2 Admin Tool functions to a copy of one or more of the panels supplied with the product.

Tip: Use the existing code in the panel that you are modifying as a template, and make the necessary changes for the new function. When you complete your modifications, change the Db2 Admin Tool source by creating an SMP/E usermod to ensure that changes are not lost if maintenance is applied to the product.

You can develop new, independent applications by using the sample application panels included with Db2 Admin Tool as templates.

Regardless of whether you are creating or extending Db2 Admin Tool applications, the process involves creating ISPF panels that specify how Db2 Admin Tool should perform SQL processing and dialog control.

Sample application

Db2 Admin Tool includes a sample application that you can use to help you create your own applications.

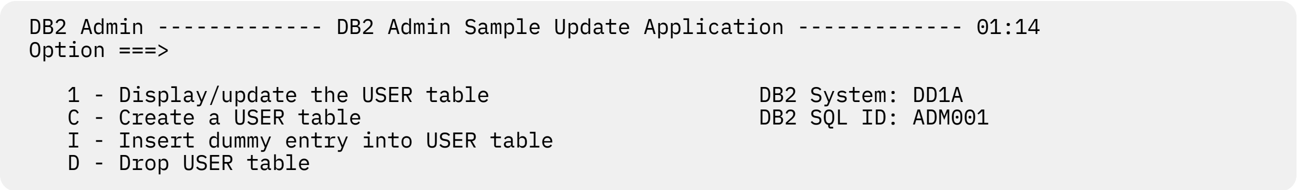
The sample application consists of three ISPF panel source members located in library SADBPLIB. Their names are ADB2S, ADB2S1, and ADB2SU. Use these sample panels as templates to create your own application.

Recommendation: To better understand the concepts in this chapter, examine these ISPF panel source members.

The sample application shows how to maintain a small DB2 table called USER. The columns in the USER table are:

```
USERID      CHAR(08) NOT NULL
EMPNAME     CHAR(30) NOT NULL
EMPLNO      CHAR(05) NOT NULL
COMMENTS    CHAR(30) NOT NULL
```

Access the sample application from the DB2 **Administration Menu** panel by specifying option S (it is not included in the list of options). The **DB2 Admin Sample Update Application** panel is displayed, as shown in the following figure.



```
DB2 Admin ------------- DB2 Admin Sample Update Application ------------- 01:14
Option ===>

   1 - Display/update the USER table                     DB2 System: DD1A
   C - Create a USER table                               DB2 SQL ID: ADM001
   I - Insert dummy entry into USER table
   D - Drop USER table
```

Figure 609. DB2 Admin Sample Update Application panel (ADB2S)

The following options are valid:

1
Displays the USER table. From this display, you can use line commands I, U, and D to insert, update, and delete rows.

C
Creates the *sqlid.USER* table (in default database DSNDB04).

I
Inserts a dummy row into the table so it is possible to display or update the table using option 1.

D
Drops the USER table.

Types of panels

You can create different types of panels with Db2 Admin Tool.

You can create the following types of panels:

Menu panels
These panels are typically at the top of a hierarchy of other panels. Menu panels specify the options that are available to the user.

Table display panels
These are ISPF table display panels on which data from DB2 or ISPF tables are displayed.

Data entry panels
On these panels, a user enters data that is input to a DB2 SQL statement, DB2 command, or DB2 Admin CLIST.

Help panels
These are standard ISPF help panels to guide the user in performing a task.

For a new application, you typically create a menu panel and a number of data entry and table display panels.

Controlling Db2 Admin Tool processing

You control Db2 Admin Tool processing by setting variables on the panels.

During processing, Db2 Admin Tool looks at the variables and then processes the related commands or statements accordingly. If no variables are set, Db2 Admin Tool redisplays the panel unchanged.

You can set the following variables on the panels:

PANEL
The name of the next panel Db2 Admin Tool should display. If this variable is used with an SQL SELECT statement, the next panel should be an ISPF table display panel that shows the rows returned by DB2. On a menu panel, set the PANEL variable to the panel name Db2 Admin Tool should display for a particular choice.

SQLSTMT
Any SQL statement that DB2 can execute. If the statement is an SQL SELECT, Db2 Admin Tool creates an intermediate ISPF table, fetches the rows, adds the rows to the ISPF table, and shows the result on the specified panel. If no panel is specified, the default table display panel is shown. Multiple SQL statements can be specified; they must be separated by a semicolon (;).

ISPFSTMT
Any ISPF statement that can be executed by the ISPEXEC ISPF API. This variable is useful for invoking your own CLISTs, EXECs, or other TSO/ISPF applications. Multiple statements can be specified; they must be separated by a semicolon (;).

DB2ACMD
Any Db2 Admin Tool primary command, which includes DB2 commands, ISPF statements, and SQL statements.

Db2 Admin Tool processing flow

After a panel is displayed, Db2 Admin Tool examines the variables and processes the instructions.

Db2 Admin Tool examines the variables and processes the instructions according to the following rules:

- If the user presses END, the previous panel is displayed.
- If variable ISPFSTMT is set, all ISPF statements are processed first.
- If variable SQLSTMT is set, the SQL statements are processed one by one. If DB2 returns rows, the result on the panel named in the variable PANEL is displayed. If the variable PANEL is not set, the default panel is displayed.
- If the variable PANEL is set, the specified panel is displayed.
- If the variable DB2ACMD is set, the Db2 Admin Tool commands are processed.

The process flow that Db2 Admin Tool follows is shown in the following figure.

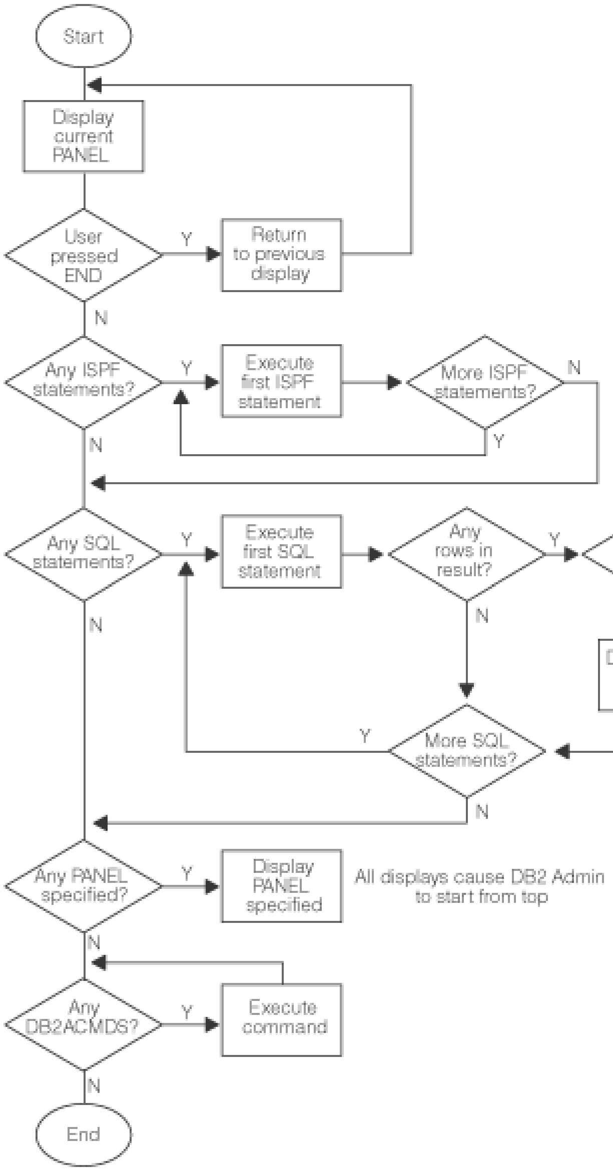


Figure 610. DB2 Admin logic diagram

Panel naming conventions

You can use Db2 Admin Tool panels as models for creating your own panels.

However, you must use a different prefix in your panel names.

Db2 Admin Tool panels have the prefix ADB. The suffix normally identifies the option that you selected to display the panel. For example, ADB1T is the panel for option 1 on the DB2 Administration Menu and option T on the following panel.

The corresponding help panels have the same name but use the prefix ADBH.

Invoking new applications

If you have created a new, independent application, you can use the Db2 Admin Tool CLIST (ADBL) to invoke it.

Use the following parameters to invoke your application:

PANEL(panel)
Name of the first panel to be shown

SYSTEM(name)
DB2 subsystem that is to be used

Example: To start a Db2 Admin Tool with your own customized panel, invoke the CLIST by issuing the following command:

```
%ADBL PANEL(yourpanel)
```

Updating rows by using SQL

If your Db2 Admin Tool application will use SQL to update rows, perform the updates on a separate panel.

Updating rows on the same panel will result in a copy of the data on the table display panel, but updated data in DB2. When you use a separate panel for updates, Db2 Admin Tool refreshes the data in the table display panel automatically when DB2 data changes.

Also, Db2 Admin Tool issues an SQL COMMIT before each display, so if you have concurrent users of your application, you probably should have a time stamp for the latest updates to rows.

If you are updating rows using SQL, consider using the structure shown in the following figure for your Db2 Admin Tool application.

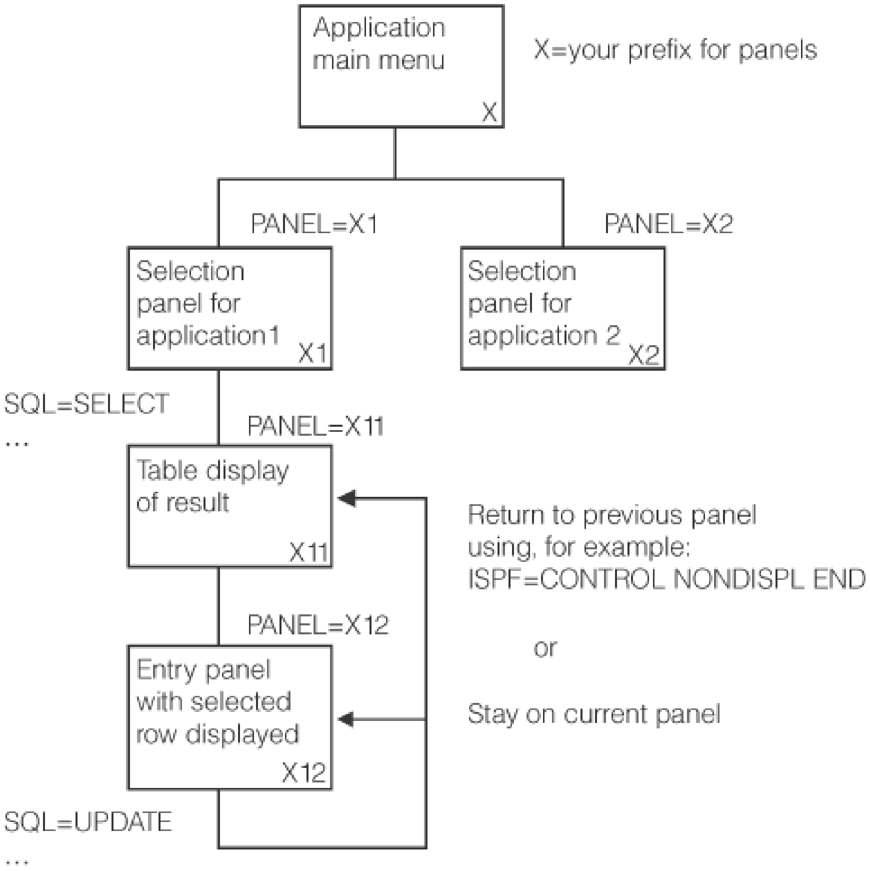

Application main menu
X
X=your prefix for panels
PANEL=X1
PANEL=X2
Selection panel for application1
X1
Selection panel for application 2
X2
SQL=SELECT
...
PANEL=X11
Table display of result
X11
Return to previous panel using, for example:
ISPF=CONTROL NONDISPL END
or
Stay on current panel
PANEL=X12
Entry panel with selected row displayed
X12
SQL=UPDATE
...

Figure 611. An example application structure

Variables in your applications

You can use two types of variables in your Db2 Admin Tool application.

- General Db2 Admin Tool variables
- Variables that contain column values, set as a result of an SQL SELECT and a line command that selected the row

All variables are located in the ISPF function pool.

General Db2 Admin Tool variables

The general Db2 Admin Tool variables are DB2SYS, DB2AUTH, and DB2AMROW.

DB2SYS
Indicates the Db2 system ID. The DB2SYS variable is set by the Db2 Admin Tool CLIST.

DB2AUTH
Indicates the current Db2 authorization ID.

DB2AMROW
Indicates the maximum number of rows to fetch. The default is 1000.

Variables that contain column values

After an SQL SELECT statement is executed, Db2 Admin Tool defines a variable for each column of the result by using the ISPF VDEFINE service.

Therefore, these variables are available to your application. When you select a row, the content of the column variables have the values for that row.

The names of column variables are the same as DB2 column names except for the following differences:

- ISPF variable names have a maximum of eight characters. If the DB2 column name exceeds eight characters, it is truncated to eight characters. For example, the DB2 column name CLUSTERTYPE has the ISPF name CLUSTERT.
- Special characters, such as underscores in DB2 column names, are replaced by the at sign (@). For example, DB2 column name EMPL_NAME has the ISPF name EMPL@NAM.
- If duplicate column names exist in the result, all but the first duplicate column name are given ISPF name DUP0001, DUP0002, and so on. For example, SELECT CREATEDBAAUTH,CREATEDBCAUTH FROM SYSIBM.SYSUSERAUTH is given ISPF names CREATEDB and DUP0001.
- All DB2 SELECT expressions are given ISPF names COL0001, COL0002, and so on. For example, SELECT CURRENT DATE is given ISPF name COL0001.
- Table search argument variables are named in the same way as ISPF names, but they are truncated to seven characters and given the prefix @. Duplicates are named @DUP0001, @DUP0002, and so on.

Copies of the Db2 catalog

You can create copies of the Db2 catalog to be used by Db2 Admin Tool.

For example, you can choose to use a different copy of the catalog for each weekday and create a backup copy that is associated with each weekday. This strategy allows you to examine previous definitions in the backup copies of the Db2 catalog. As another example, you might allow only the system administrator to examine the active Db2 catalog and allow developers to access only a copy of the Db2 catalog. This strategy can result in decreased contention on the catalog that can be caused by the developers' queries, while still allowing the system administrator to maintain the active Db2 catalog.

Db2 Admin Tool uses the *catalog copy version table*, ADBCATVT, to keep track of the available Db2 catalog copies. You are responsible for adding an entry to this table for each copy of the catalog that you want to use. For information about how to create catalog copies, including how to add an entry to the ADBCATVT table, see “Making copies of the Db2 catalog for Db2 Admin Tool” on page 1049.

When you select any options on the **System Catalog (ADB21)** panel, you can choose to use a copy of the Db2 catalog. For instructions on how to select the copy that you want to use, see “Switching copies of the Db2 catalog” on page 268.

You can also choose to use the catalog of a remote Db2 subsystem. This method for using a remote site catalog is different from the method that is provided by the **DD** option (distributed Db2 systems) on the **DB2 Administration Menu (ADB2)** panel. (For more information about that option, see “Db2 Admin Tool distributed support” on page 1054.) A GEN request can use a remote copy of the catalog if you use a catalog alias (the catalog copy type is A) and specify the alias' location for routing to the remote site.

Making copies of the Db2 catalog for Db2 Admin Tool

If you do not want your Db2 Admin Tool users to run queries against the actual Db2 catalog, you can use one or more local copies of the catalog. Using a local copy of the catalog can help avoid any performance impact to your production Db2 subsystem if you run complex catalog searches. Using local copies also enables you to work with different versions of the catalog, so that you can perform certain actions such regenerating the DDL of an object that was accidentally dropped.

Before you begin

Recommendation: Run the RUNSTATS utility on the catalog table spaces before creating copies of the catalog. These updated statistics help Db2 Admin Tool calculate reasonable space requirements for copies of the catalog table spaces when it builds CREATE TABLESPACE and CREATE INDEX statements.

About this task

To make a copy of the Db2 catalog, you must first add an entry to the catalog copy version table (ADBCATVT) and then generate the jobs to make the copy. This procedure guides you through that process.

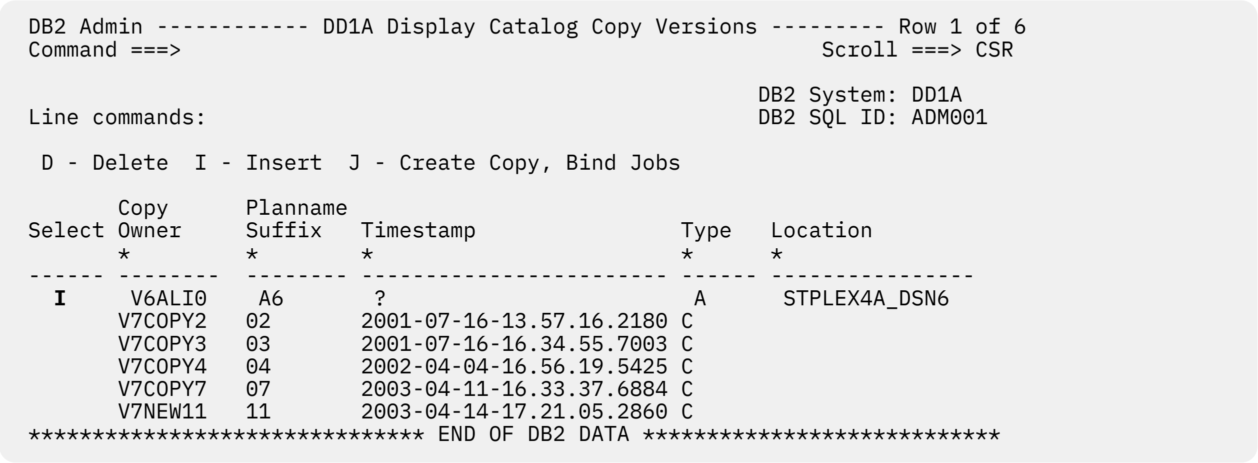
Procedure

To make copies of the Db2 catalog for Db2 Admin Tool:

1. On the **DB2 Administration Menu (ADB2)** panel, specify the CC option, and press Enter.

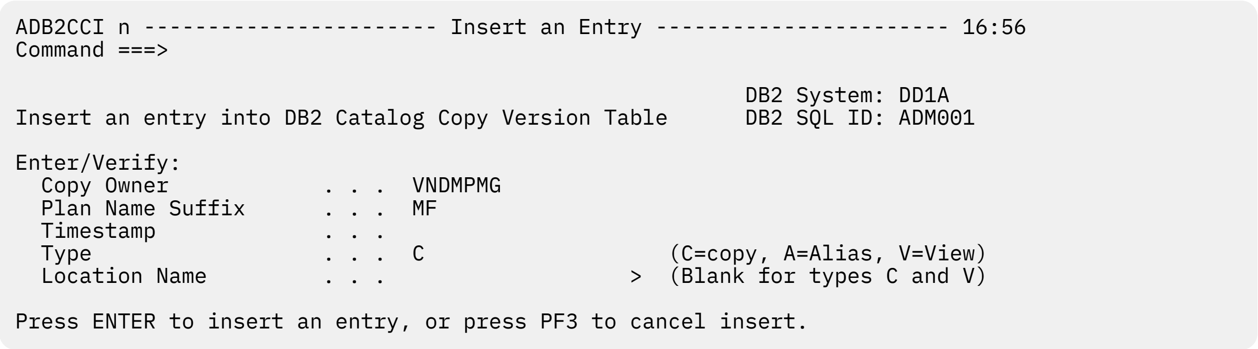
   If the catalog copy version table, ADBCATVT, is populated, the **Display Catalog Copy Versions (ADB2CCD)** panel is displayed. If this table does not have any rows, the **Insert an Entry (ADB2CCI)** panel is displayed.

2. If **Display Catalog Copy Versions (ADB2CCD)** panel is displayed, specify I next to any row, and press Enter:



```
DB2 Admin ------------ DD1A Display Catalog Copy Versions --------- Row 1 of 6
Command ===>                                                   Scroll ===> CSR

                                                          DB2 System: DD1A
Line commands:                                            DB2 SQL ID: ADM001

 D - Delete  I - Insert  J - Create Copy, Bind Jobs

       Copy      Planname
Select Owner     Suffix   Timestamp                  Type   Location
       *         *        *                          *      *
------ --------  -------- -------------------------- ------ ----------------
  I     V6ALI0    A6       ?                          A      STPLEX4A_DSN6
       V7COPY2   02       2001-07-16-13.57.16.2180 C
       V7COPY3   03       2001-07-16-16.34.55.7003 C
       V7COPY4   04       2002-04-04-16.56.19.5425 C
       V7COPY7   07       2003-04-11-16.33.37.6884 C
       V7NEW11   11       2003-04-14-17.21.05.2860 C
******************************* END OF DB2 DATA ****************************
```

3. On the **Insert an Entry (ADB2CCI)** panel, specify the following information for the catalog copy that you want to make, and press Enter:



```
ADB2CCI n ----------------------- Insert an Entry ----------------------- 16:56
Command ===>

                                                          DB2 System: DD1A
Insert an entry into DB2 Catalog Copy Version Table      DB2 SQL ID: ADM001

Enter/Verify:
  Copy Owner            . . .  VNDMPMG
  Plan Name Suffix      . . .  MF
  Timestamp             . . .
  Type                  . . .  C                    (C=copy, A=Alias, V=View)
  Location Name         . . .                  >  (Blank for types C and V)

Press ENTER to insert an entry, or press PF3 to cancel insert.
```

Copy Owner
   The Db2 authorization ID of the creator of the copy. This field must be unique within the catalog copy version table.

Plan Name Suffix
   The suffix that is appended to the name of certain plans and the collection that are bound with the **Copy Owner** qualifier. This value can be any two characters. This field must be unique within the catalog copy version table.

Type
   The type of catalog copy:

   C
      A copy of the local Db2 catalog.

A

An alias for a remote Db2 catalog.

V

A view against a Db2 catalog.

Location Name

The remote location. This field applies only if **Type** is A.

After you press Enter, if the **Statement Execution Prompt (ADB2PSTM)** panel is displayed, select the appropriate option to run the statement.

The entry is added to the catalog copy version table, ADVCATVT.

4. Exit back to the **Display Catalog Copy Versions (ADB2CCD)** panel.
5. Specify the J line command next to the new entry, and press Enter. This command generates create, bind, and copy jobs to copy the catalog.
6. On the **Create Catalog Copy and Bind Batch Jobs (ADB2CCJ)** panel, specify the required fields, and press Enter to generate the jobs:

```
ADB2CCJ n ---------DD1A Create Catalog Copy and Bind Batch Jobs -------- 16:56
Command ===>


Specify the following for DB2 Admin CATALOG COPY:       DB2 System: DD1A
                                                        DB2 SQL ID: ADM001

Catalog Copy Owner . . : VNDMPMG
Planname Suffix  . . . : VN
Type . . . . . . . . . : C (C=copy, A=Alias, V=View)

Data set information:
  PDS for jobs  . . . . . . . .'VNDMPM2.TEST.JCL1'
  Prefix for work data sets . . VNDMPM2

Catalog Copy options (for type C only):
  Database Name   . . . . . . . VNDMPM2     (? to look up existing database)
  Storage Group Name  . . . . . VNDMPM2  >  (? to look up existing stogroup)
  Run SQLID . . . . . . . . . . VNDMPM2  >
  Catalog Copy Method . . . . . L           (L=LOAD from Cursor, U=UNLOAD/LOAD)
  Grant table privilege to  . . VNDM001
Miscellaneous options:
  Batch job PDS unit type . . . SYSDA
  Work data set unit type . . . SYSDA

BP - Change batch job parameters
UO - Customize utility options
```

- For all types, specify the name of an existing partitioned data set (PDS) where the jobs are to be generated.
- For type C (copies of a local Db2 catalog), specify the following information:
  - The name of the database and the storage group for the table space that Db2 Admin Tool is to create to contain the new tables. These new tables are created with the CREATE TABLE LIKE statement based on the Db2 catalog tables. The name of the table space that is created to contain these new tables is the same as the qualifier of the copy (the value that you specified in the **Copy Owner** field on the **Insert an Entry (ADB2CCI)** panel).
  - The method to use to copy the Db2 catalog to the new tables:

    UNLOAD/LOAD

    This method unloads data into data sets as one process and loads the data into the new tables as a second process by using the UNLOAD and LOAD utilities. The generated CPYRUN*xx* job uses TEMPLATE utility statements to define output and work data sets. You can modify those TEMPLATE statements as needed. (This function does not use any user-specified templates.)

    You can manually estimate the space on the Db2 UNLOAD and LOAD templates, UNLDDN and INDDN, by using the last image copy on tape. To estimate the size of the unload data set, use the information from COPYPAGESF or NPAGESF in the SYSIBM.SYSCOPY catalog table.

Manually add the SPACE option in the UNLOAD steps of your JCL for the templates that are associated with INDDN and UNLDDN.

The following example shows an ULOAD utility template with the space parameters specified. This example shows the required options if the image copy is on tape.

```
TEMPLATE ADBCDATA DSN '<dsnprefix>.ADBCCPJ1.DSNXSR.SYSXSR'
UNIT SYSDA SPACE(10,50) CYL;
```

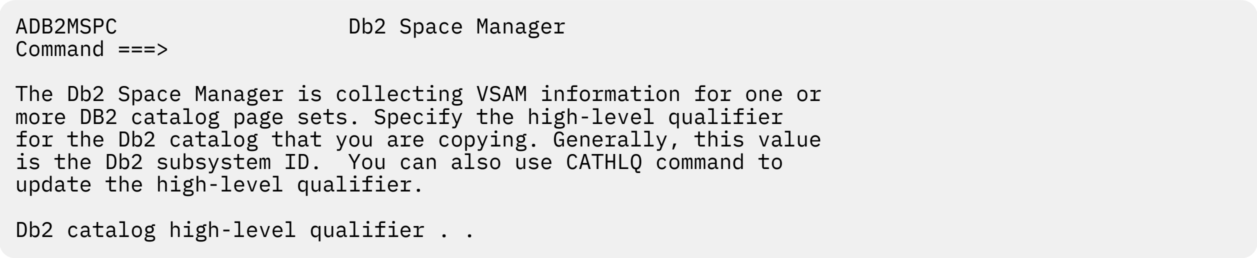
Related information:

Syntax and options of the TEMPLATE control statement (Db2 12 for z/OS)

LOAD from Cursor

This method is the default. Use this method if you do not need to move or modify the catalog data outside of this process. This method reduces the I/O load of the entire process and requires no work data sets. When you choose **LOAD from Cursor**, Db2 Admin Tool still uses the **UNLOAD/LOAD** method on catalog tables that contain LOB columns. Catalog tables that contain LOB columns also contain columns that are defined as GENERATED ALWAYS, and Db2 does not allow GENERATED ALWAYS columns in the specification list when the **LOAD from Cursor** method is used. Therefore, the **UNLOAD/LOAD** method must be used. LOB columns are unloaded to a VBS data set.

After you press Enter, if the **DB2 Space Manager (ADB2MSPC)** panel is displayed, specify the high-level qualifier for the Db2 catalog that you are copying and press Enter. This value is generally the Db2 subsystem ID.



```
ADB2MSPC                  Db2 Space Manager
Command ===>

The Db2 Space Manager is collecting VSAM information for one or
more DB2 catalog page sets. Specify the high-level qualifier
for the Db2 catalog that you are copying. Generally, this value
is the Db2 subsystem ID.  You can also use CATHLQ command to
update the high-level qualifier.

Db2 catalog high-level qualifier . .
```

The following jobs are generated: (In all of these jobs, *xx* is the plan name suffix.)

- For type A (aliases of a distributed Db2 catalog), one job is created: ALIBND*xx*. This job creates aliases for the Db2 catalog tables of the distributed subsystem at the given location. This job also binds the plans that Db2 Admin Tool needs to access the aliases.
- For type C (copies of a local Db2 catalog), the following two jobs are created:
  - DDLBND*xx* is the create and bind job. It creates tables with the CREATE TABLE LIKE statement based on the Db2 catalog tables and binds the plans. Run this job once to create all the tables for the copy of the catalog and to bind the plans that Db2 Admin Tool uses when this copy is selected.
  - CPYRUN*xx* refreshes the copy. Run this job to create the initial copy of the tables; rerun it whenever the copy needs to be refreshed. This job also runs the RUNSTATS utility against the table space that contains the copy and updates the timestamp field of the catalog copy version record.
- For type V (views of a local Db2 catalog), one job is created: VIEBND*xx*. This job creates views for the local catalog tables. You can modify this job to add predicates to the views to limit which rows are accessible to users. However, restricting which rows are accessible can affect the ability of Db2 Admin Tool to retrieve information and thus result in incomplete information being returned.

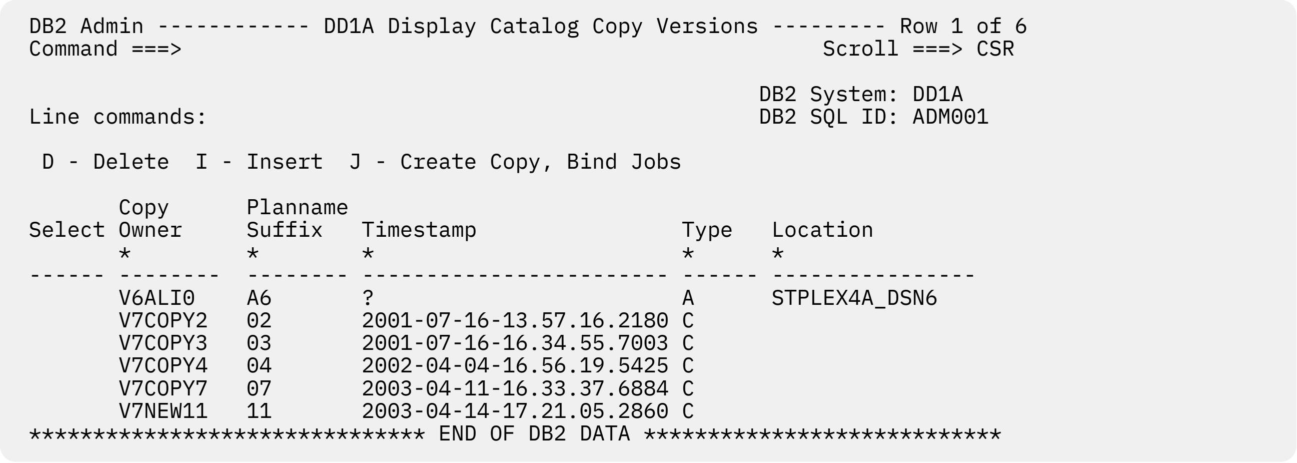
7. Run the appropriate job or jobs.

   Db2 Admin Tool renames any duplicate indexes that are created during catalog copy processing. For the new names of the duplicate indexes, see step ISPFBAT in the generated job.

   Tip: The catalog copy process includes building tables that match the names of the catalog tables. Also, indexes are built for those tables that match the names of the current set of indexes on the catalog tables. Tables and indexes with the same qualifier and name might already exist as objects other than the intended catalog copy objects. If a duplicate object exists, SQLCODE -601 is issued when you run the DDL to create the new catalog copy. If you receive this error, you need to modify the DDL and restart the step. Db2 Admin Tool detects certain duplicate index errors when creating the DDL for the index and attempts to avoid the error by creating a new name for the index that is based on the old name. However, Db2 Admin Tool cannot detect and handle all cases.

Display Catalog Copy Versions (ADB2CCD) panel

The **Display Catalog Copy Versions (ADB2CCD)** panel lists the Db2 catalog copies that are available to be used by Db2 Admin Tool.



```
DB2 Admin ------------ DD1A Display Catalog Copy Versions --------- Row 1 of 6
Command ===>                                                  Scroll ===> CSR

                                                         DB2 System: DD1A
Line commands:                                           DB2 SQL ID: ADM001

 D - Delete  I - Insert  J - Create Copy, Bind Jobs

       Copy      Planname
Select Owner     Suffix   Timestamp                  Type   Location
       *         *        *                          *      *
------ --------  -------- -------------------------- ------ ----------------
       V6ALI0    A6       ?                          A      STPLEX4A_DSN6
       V7COPY2   02       2001-07-16-13.57.16.2180 C
       V7COPY3   03       2001-07-16-16.34.55.7003 C
       V7COPY4   04       2002-04-04-16.56.19.5425 C
       V7COPY7   07       2003-04-11-16.33.37.6884 C
       V7NEW11   11       2003-04-14-17.21.05.2860 C
****************************** END OF DB2 DATA ****************************
```

Figure 612. ***Display Catalog Copy Versions (ADB2CCD)*** *panel*

This panel includes the following columns:

Select
An input field where you can enter a line command.

Copy Owner
The Db2 authorization ID of the creator of the copy. This field must be unique within the table.

Planname Suffix
The suffix that is appended to the name of certain plans and the collection that are bound with the **Copy Owner** qualifier. This value can be any two characters. This field must be unique within the table.

Timestamp
The time when the copy of the catalog was last refreshed. When inserting an entry, leave this field blank.

Type
The type of catalog copy:

C
A copy of the local Db2 catalog. When creating an entry for a copy of a local catalog, enter the copy owner, plan name suffix, and type.

A
An alias for a catalog at a remote site. When creating an entry for a remote catalog, enter the copy owner, plan name suffix, type, and location of the remote catalog.

V
A view of a local Db2 catalog. When creating an entry for a view of a local catalog, enter the copy owner, plan name suffix, and type.

Location Name
The location of a remote catalog. This field applies only if **Type** is A.

This panel supports the following line commands:

D

Delete a catalog copy entry from the table.

I

Insert a new catalog copy entry into the table.

J

Generate create, bind, and copy jobs to copy the Db2 catalog. The jobs that Db2 Admin Tool generates are decribed in "Making copies of the Db2 catalog for Db2 Admin Tool" on page 1049.

Using previously defined copies of the Db2 catalog

If your installation defined copies of the Db2 catalog before migrating to a new release of Db2 Admin Tool or a new Db2 version or function level, you need to perform an additional step after migrating.

Procedure

To use previously defined copies of the Db2 catalog, you must update the catalog copy tables by completing one of the following procedures:

- **Procedure 1: Regenerate the catalog copies**

  a) On the **DB2 Administration Menu (ADB2)** panel, specify the CC option, and press Enter.

  The **Display Catalog Copy Versions (ADB2CCD)** panel is displayed. If this panel Is not displayed, no catalog copies have been defined.

  b) Issue the J line command for each entry to regenerate the jobs for the new release of Db2 Admin Tool.

  c) In all DDLBND*xx* and ALIBND*xx* jobs, change the second line of the job as follows:

  Existing line: `//* RESTART=stepname, <=== For restart, remove * and enter step name`

  Change the line to:`// RESTART=BIND`

  d) Run the BIND step of all DDLBND*xx* and ALIBND*xx* jobs.

- **Procedure 2: Drop and recreate the catalog copies**

  Regenerate the catalog copy job DDLBND*xx* and submit it. This job will sync your copy catalog tables with the latest Db2 catalog, so that any new columns are included. Upon successful completion of the job, submit the copy job CPYRUN*xx*.

- **Procedure 3: Use Db2 Object Comparison Tool to update the existing catalog copies**

  Use the Db2 catalog as your source and the catalog copy as your target. This comparison should generate the ALTER TABLE ADD COLUMN statements needed to make sure that catalog copy tables match the Db2 catalog tables. For detailed instructions on how to compare these objects and apply the changes, see Comparing Db2 objects (IBM Db2 Object Comparison Tool for z/OS 13.1.0).

Db2 Admin Tool distributed support

You can use Db2 Admin Tool to connect to remote systems.

With Db2 Admin Tool distributed support, you can perform the following actions:

- Build utility jobs and submit them to run on remote systems.
- Perform alter and migrate functions for remote systems.
- Issue SQL statements against remote systems.
- Issue distributed GRANT and REVOKE commands.
- Issue other commands on remote systems.

Db2 Admin Tool uses an SQL CONNECT STATEMENT to connect to a remote system. Db2 Admin Tool can also use three-part names (for example, *location.schema.name*) where Db2 performs an implicit connect to the remote system.

Setup for distributed access

To make Db2 Admin Tool and Db2 Object Comparison Tool available on remote subsystems you might need to issue additional GRANT statements on the packages at the remote site. Whether this action is required depends on your specific DRDA setup. If needed, these GRANT statements (to grant privileges to the owner of the plan at the local site) run on the packages after outbound and inbound AUTHID translation is done. This process is the same process that you use to set up SPUFI to run on a remote system.

The DRDA setup can be different between installations and sometimes within one installation. Some installations have policies that also restrict the use of GRANT to PUBLIC.

If you want to use the DDF functions in Db2 Admin Tool 12.1 to access a remote system and Db2 Admin Tool is not customized on the remote system that you want to access, you must bind the Db2 Admin Tool 12.1 packages on the remote system. You can take one of the following actions to bind these packages:

- Issue the following BIND command:

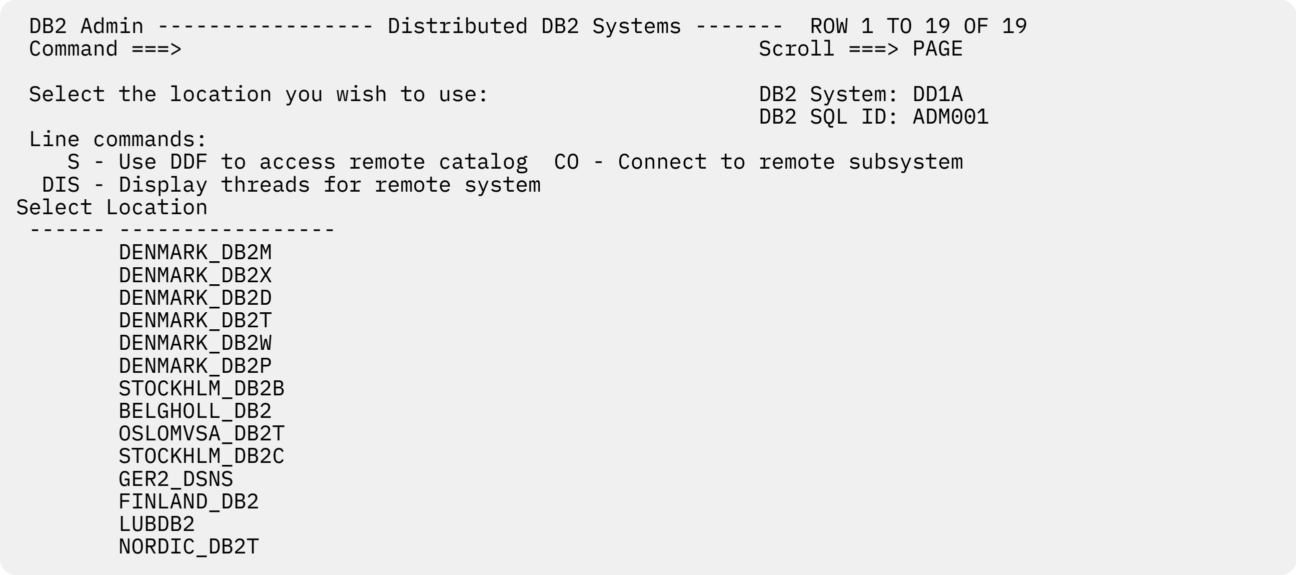
  ```
  BIND PACKAGE(<remotelocation>.ADBL) COPY(ADBL.<package>) COPYVER(<version>)
  ```

  This command copies and binds the package from the local system to the remote system.
- On the **Packages (ADB21K)** panel, use the BC line command on the package that you want to copy and bind.

If you need to invoke additional functions, such as GEN, on a remote system on which Db2 Admin Tool is not installed, you must also bind the packages for those functions on the remote system. For example, GEN requires the ADB2RE* and other packages to be bound on that system. If many functions need to be used, modify and run the TCz ADBBIND job on the remote system.

Distributed DB2 Systems (ADB2DDF) panel

To use Db2 Admin Tool distributed support, select option DD from the **DB2 Administration Menu (ADB2)** panel. This option displays the **Distributed DB2 Systems (ADB2DDF)** panel:



```
 DB2 Admin ----------------- Distributed DB2 Systems -------  ROW 1 TO 19 OF 19
 Command ===>                                             Scroll ===> PAGE

 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
 ------ -----------------
        DENMARK_DB2M
        DENMARK_DB2X
        DENMARK_DB2D
        DENMARK_DB2T
        DENMARK_DB2W
        DENMARK_DB2P
        STOCKHLM_DB2B
        BELGHOLL_DB2
        OSLOMVSA_DB2T
        STOCKHLM_DB2C
        GER2_DSNS
        FINLAND_DB2
        LUBDB2
        NORDIC_DB2T
```

*Figure 613. **Distributed DB2 Systems (ADB2DDF)** panel*

This panel lists the remote Db2 subsystems that are available from the current Db2 subsystem, or *local subsystem*. You can perform the following actions by using the specified line command:

DIS
Displays the active threads for the location or system you select.

S

Selects the remote subsystem for which you want to access the remote system catalog.

CO

Connects directly to a remote subsystem for issuing remote requests.

You can also use the CONNECT `location_name` primary command to connect to a remote subsystem.

Restrictions when connecting to a remote subsystem

When using the **DD - Distributed DB2 systems** function to connect to a remote subsystem, the following actions are restricted:

- When accessing a remote Db2 system catalog, some functions in the Db2 Admin Tool system catalog dialog are disabled. For example, you cannot issue Db2 DISPLAY or GEN commands. Also, unless prompting is on, you also cannot issue Db2 BIND, REBIND, or FREE commands.
- If you connect to a remote subsystem that does not have an entry in the ADBTPARM customization table, alter, migrate, and utility jobs are not allowed, and an error message is displayed. (The Db2 subsystem parameters are stored in ISPF table member ADBTPARM in the ISPTLIB table library that is specified in TCz.)
- You cannot use option 1 of the Space Management function (display page set space by database).
- You cannot issue SM line commands on the database and table space panels.
- You cannot interface to other Db2 products from a remote subsystem.

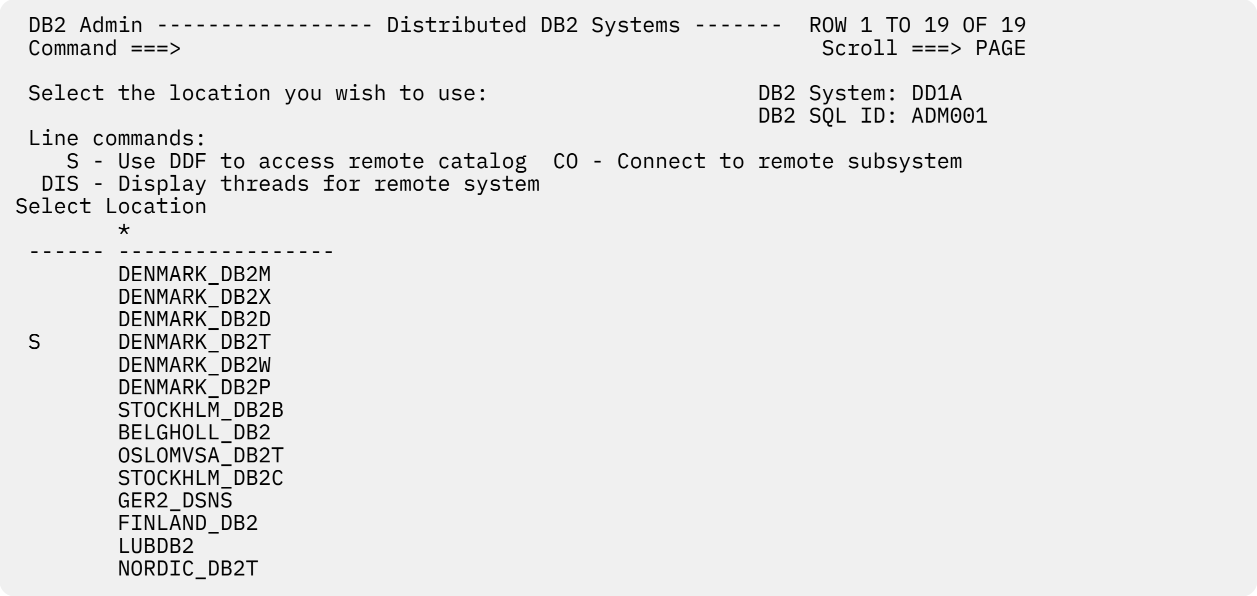
Additionally, to use copies of the system catalog of a remote subsystem, the local subsystem customization must specify the owner of the catalog copy version table.

Accessing a remote subsystem

You can use Db2 Admin Tool to access the catalog on a remote Db2 subsystem.

Procedure

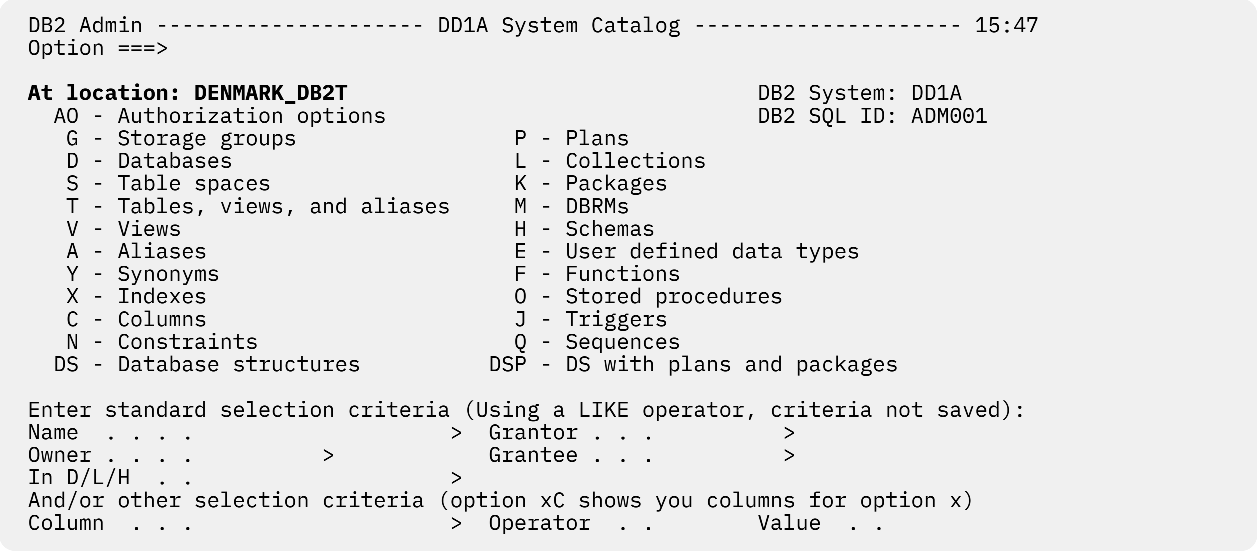
1. On the **DB2 Administration Menu (ADB2)** panel, specify option DD, and press Enter.
2. On the **Distributed DB2 Systems (ADB2DDF)** panel, issue the S line command next to the remote Db2 subsystem for which you want to access the catalog.



```
 DB2 Admin ----------------- Distributed DB2 Systems -------  ROW 1 TO 19 OF 19
 Command ===>                                                  Scroll ===> PAGE

 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
        *
 ------ -----------------
        DENMARK_DB2M
        DENMARK_DB2X
        DENMARK_DB2D
 S      DENMARK_DB2T
        DENMARK_DB2W
        DENMARK_DB2P
        STOCKHLM_DB2B
        BELGHOLL_DB2
        OSLOMVSA_DB2T
        STOCKHLM_DB2C
        GER2_DSNS
        FINLAND_DB2
        LUBDB2
        NORDIC_DB2T
```

*Figure 614. **Distributed DB2 Systems (ADB2DDF)** panel*

The **System Catalog (ADB21)** panel is displayed for the selected location: The release level and mode of your Db2 subsystem affect the options that are available. All generated batch utility jobs, ALTER commands, and MIGRATE commands are sent to the remote subsystem (or the target system for the migrate jobs) for execution after the jobs have been submitted on the local subsystem.



```
DB2 Admin --------------------- DD1A System Catalog --------------------- 15:47
Option ===>

At location: DENMARK_DB2T                                DB2 System: DD1A
  AO - Authorization options                             DB2 SQL ID: ADM001
   G - Storage groups                  P - Plans
   D - Databases                       L - Collections
   S - Table spaces                    K - Packages
   T - Tables, views, and aliases      M - DBRMs
   V - Views                           H - Schemas
   A - Aliases                         E - User defined data types
   Y - Synonyms                        F - Functions
   X - Indexes                         O - Stored procedures
   C - Columns                         J - Triggers
   N - Constraints                     Q - Sequences
  DS - Database structures           DSP - DS with plans and packages

Enter standard selection criteria (Using a LIKE operator, criteria not saved):
Name  . . . .                    >  Grantor . . .          >
Owner . . . .          >            Grantee . . .          >
In D/L/H  . .                    >
And/or other selection criteria (option xC shows you columns for option x)
Column  . . .                    >  Operator  . .        Value  . .
```

*Figure 615. **System Catalog (ADB21)** panel for a remote subsystem*

3. After connecting to the remote subsystem, issue `OPTIONS` BP to set up JOB cards for the remote subsystem. The last JOB card that is used remains active until another BP command is issued. If you have not set up a JOB card for the remote subsystem, the JOB cards for the local subsystem are used on the remote subsystem.

Chapter 7. Troubleshooting

Use these topics to diagnose and correct problems that you experience with Db2 Admin Tool.

For help troubleshooting customization problems with Tools Customizer, see Tools Customizer troubleshooting (IBM Tools Customizer for z/OS 1.1).

Gathering diagnostic information

Before you report a problem with Db2 Admin Tool to IBM Software Support, you need to gather the appropriate diagnostic information.

If you receive Db2 Admin Tool error messages that do not contain adequate information regarding the actions you should take, use the following information to diagnose common problems before contacting the IBM Software Support. The information that you gather to diagnose the problem is required when you open an incident with the Db2 Admin Tool Support team.

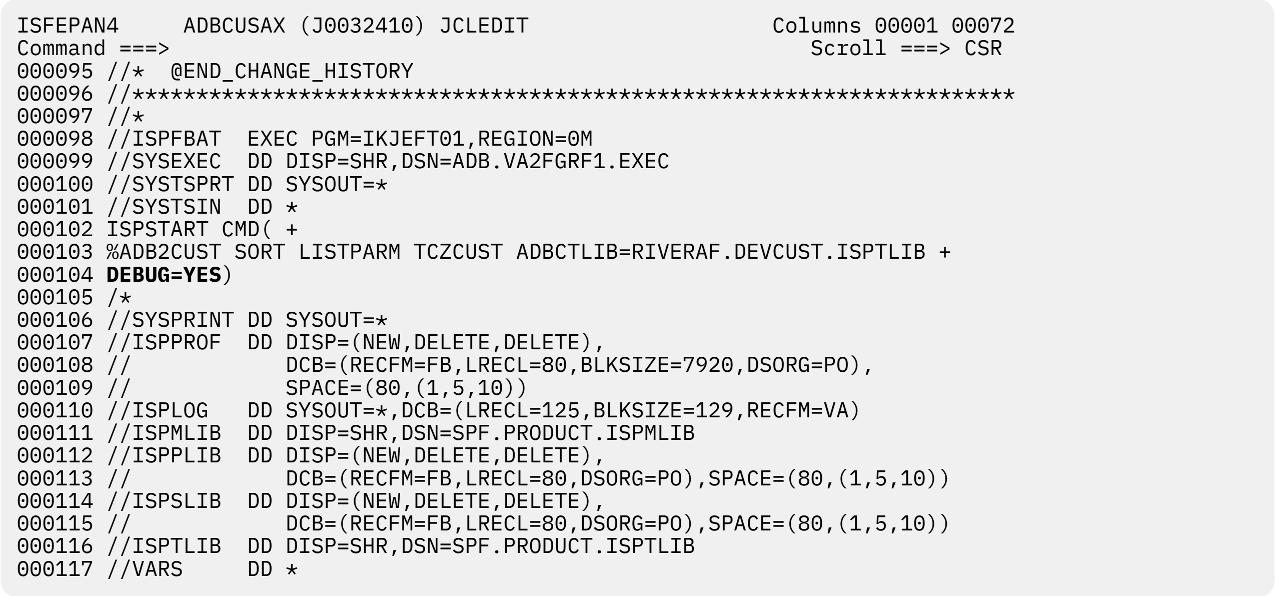
- For general abends, obtain the following information:
  - ABEND code
  - Dump title
  - Failing module/CSECT name
  - A printout of the traceback from a Language Environment (LE) dump
  - Recent maintenance applied
  - Recent changes to the system
  - Frequency of the abend, or prevailing conditions when the abend occurred. For example, does the abend occur for only a single user ID?
  - VTAM message
  - MVS ABENDs
  - Dumps, as appropriate
- When contacting the support team, provide the following required documentation :
  - Whether Db2 data sharing was used
  - Whether a remote Db2 subsystem was involved
  - A complete explanation of the problem encountered
  - Complete job output of failing jobs
  - If problems occur when using online mode, screen shots of any error messages and screen shots of all panels prior to the error
  - Appropriate input parameters for re-creating the problem scenario
  - Complete DDL to define any objects involved with the error, if appropriate
  - The *userid*.ADBEDIAG.REPORT data set

    To create this data set, issue the command `TSO ADBEDIAG REPORT`. Among other things, this report includes the following information:

    - Db2 Admin Tool version number, release number, and maintenance level
    - Db2 version number, release number, and maintenance level
    - A screen shot of the **DB2 Admin Options (ADB2P)** panel
  - Any work statement lists (WSLs), mask data sets, or ignore data sets that apply.
  - The Tools Customizer trace data set (*tso profile PREFIX*.CCQ.TRACE or *user_id*.CCQ.TRACE if TSO PROFILE NOPREFIX is used.)

- If using Change Management (CM) or CM batch, the version files, JCL and complete job output from all steps

- When troubleshooting the general customization job ADBCUST with IBM, add the DEBUG=YES parameter as shown in the following figure. This parameter produces trace information that can be shared with IBM for further analysis.



```
ISFEPAN4     ADBCUSAX (J0032410) JCLEDIT                   Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
000095 //*  @END_CHANGE_HISTORY
000096 //*********************************************************************
000097 //*
000098 //ISPFBAT  EXEC PGM=IKJEFT01,REGION=0M
000099 //SYSEXEC  DD DISP=SHR,DSN=ADB.VA2FGRF1.EXEC
000100 //SYSTSPRT DD SYSOUT=*
000101 //SYSTSIN  DD *
000102 ISPSTART CMD( +
000103 %ADB2CUST SORT LISTPARM TCZCUST ADBCTLIB=RIVERAF.DEVCUST.ISPTLIB +
000104 DEBUG=YES)
000105 /*
000106 //SYSPRINT DD SYSOUT=*
000107 //ISPPROF  DD DISP=(NEW,DELETE,DELETE),
000108 //           DCB=(RECFM=FB,LRECL=80,BLKSIZE=7920,DSORG=PO),
000109 //           SPACE=(80,(1,5,10))
000110 //ISPLOG   DD SYSOUT=*,DCB=(LRECL=125,BLKSIZE=129,RECFM=VA)
000111 //ISPMLIB  DD DISP=SHR,DSN=SPF.PRODUCT.ISPMLIB
000112 //ISPPLIB  DD DISP=(NEW,DELETE,DELETE),
000113 //           DCB=(RECFM=FB,LRECL=80,DSORG=PO),SPACE=(80,(1,5,10))
000114 //ISPSLIB  DD DISP=(NEW,DELETE,DELETE),
000115 //           DCB=(RECFM=FB,LRECL=80,DSORG=PO),SPACE=(80,(1,5,10))
000116 //ISPTLIB  DD DISP=SHR,DSN=SPF.PRODUCT.ISPTLIB
000117 //VARS     DD *
```

Figure 616. General Customization job ADBCUST with DEBUG=YES parameter added

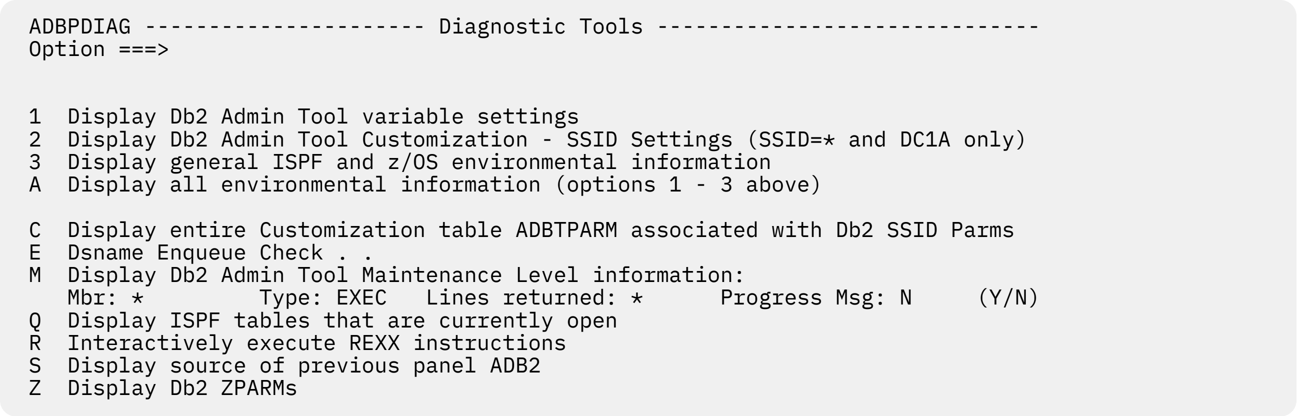
ADBEDIAG

ADBEDIAG is a Db2 Admin Tool diagnostic tool that you can use to easily gather information for IBM Software Support.

To use ADBEDIAG, enter the following command:

```
TSO ADBEDIAG
```

The **Diagnostic Tools (ADBPDIAG) panel** is displayed:



```
ADBPDIAG ---------------------- Diagnostic Tools ------------------------------
Option ===>


1  Display Db2 Admin Tool variable settings
2  Display Db2 Admin Tool Customization - SSID Settings (SSID=* and DC1A only)
3  Display general ISPF and z/OS environmental information
A  Display all environmental information (options 1 - 3 above)

C  Display entire Customization table ADBTPARM associated with Db2 SSID Parms
E  Dsname Enqueue Check . .
M  Display Db2 Admin Tool Maintenance Level information:
   Mbr: *         Type: EXEC   Lines returned: *      Progress Msg: N     (Y/N)
Q  Display ISPF tables that are currently open
R  Interactively execute REXX instructions
S  Display source of previous panel ADB2
Z  Display Db2 ZPARMs
```

*Figure 617. **Diagnostic Tools (ADBPDIAG) panel***

From this panel, you can request various diagnostic information. For detailed information about each of the options, enter HELP on the command line.

Option A is equivalent to entering `TSO ADBEDIAG REPORT` on the command line. The resulting report contains the diagnostic information that is requested by IBM Software Support. Several of the other options also have equivalent commands that you can use to invoke them. Those commands are noted in the panel help.

TSO ISRDDN

You might receive a panel message that directs you to TSO ISRDDN.

This message is as follows:

```
Analysis ended with return code = 12.  Use TSO ISRDDN to check the
file contents.  The files remain allocated and should be freed manually.
```

If you receive this message, try the following procedures resolve it:

1. First, try to exit (PF3) the procedure and then try the procedure again. Sometimes exiting the procedure releases an existing process that conflicts with the process you are trying to complete.
2. Next, access TSO and enter the command `TSO ISRDDN`. The command displays a list of files that are currently allocated in the system. Review the file list to see if you can ascertain which file might be in conflict with your procedure.

Frequently asked questions

Find answers to common questions and solutions to common problems.

Customizing Db2 Admin Tool

1. When I customize Db2 Admin Tool with the Tools Customizer panels, how can I display help information for the input fields?

   Place the cursor in the input field and press PF1.
2. What value should I specify in the **Customized Table Library** field, which is on the **Product Parameters** panel.

   If you use the Discover EXEC, specify the same dataset as the one in the **Target Table Library** field.
3. When an input field has the ">" sign and I have a long dataset name, how do I enter the name?

   You can use the EXPAND function to bring up a new panel with a greater field length.
4. Why can’t I enter input into a parameter field?

   The field is not editable or available.

   Ensure that the necessary tasks and steps are enabled first.
5. On the **Product Parameters** panel, when I enable Tasks and Steps, how can I keep the panel from scrolling back to the beginning?

   Place the cursor on the Task/Step you just enabled, and then press Enter. The panel scrolls to the current position.
6. When regenerating customization jobs, do I need to resubmit all jobs?

   When generating customization jobs for first the first time, submit the jobs. However, when you regenerate jobs, you only need to submit the jobs that contain a change.
7. Before calling other products such as Table Editor, and Cloning Tool from Db2 Admin Tool, do I need to customize these other products first?

   Yes, if the products are customizable by TCz.

Troubleshooting: The Launchpad panel is missing product settings

If the Discover EXEC is not used during an Db2 Admin Tool upgrade, the Launchpad panel (if used) might be missing Launchpad product settings that were established before the upgrade.

Symptoms

After upgrading Db2 Admin Tool, the **Launchpad Table (ADBDMT)** panel is missing product settings.

Resolving the problem

Ideally, use the Discover EXEC during the upgrade.

Alternatively, you can manually copy the ISPF table member ADBDMT from the data set that is specified in the following input field on the **Product Parameters (CCQPPRD)** panel in TCz:

```
*Customized Table lib . . . ___________________________
```

Related tasks

"Discovering Db2 Admin Tool information automatically" on page 99
You can use the Db2 Admin Tool Discover EXEC to discover information from a previous or current customization of Db2 Admin Tool.

Troubleshooting: Authorization error when running GEN or DDL

If the GEN or DDL line command fails with a message about insufficient privileges on a storage group (STOGROUP), you might need to rebind the Db2 Admin Tool package ADB2RET.

Symptoms

When running GEN or DDL against a Db2 object, the load module ADB2GEN fails with message ADB1223E and DSNT408I:

```
ADB1223E ADB2RET:  Unexpected sqlcode in : Create index-name
DSNT408I SQLCODE = -551, ERROR:  ADB DOES NOT HAVE THE PRIVILEGE TO PERFORM OPERATION CREATE
INDEX ON OBJECT stogroup-name
```

index-name is the name of the index on a global temporary table.

stogroup-name is the name of the STOGROUP.

This message indicates that the user of Db2 Admin Tool does not have sufficient authority to create an index on a global temporary table in the Db2 work file database that is created with STOGROUP *stogroup-name*. (The work file database stores global temporary tables and their indexes as well as other temporary objects and files.)

Resolving the problem

1. Check the name of the STOGROUP that is used by the work file database in your Db2 subsystem. In a non-data-sharing environment, the Db2 installation default for the work file database is DSNDB07, and the default STOGROUP for the work file database is SYSDEFLT.
2. Check whether the Db2 Admin Tool user has sufficient authority to create objects in that STOGROUP.
3. If the user does not have authority to create objects in the STOGROUP, either grant this authority or rebind the package ADB2RET with a package owner who has SYSADM authority.

Chapter 8. Db2 Admin Tool messages and codes

Use the information in these messages to help you diagnose and solve Db2 Admin Tool problems.

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).

Related concepts

"Changed messages" on page 64
Occasionally, message text and suffix values might change. If you have code that checks for message text or numbers, use the list of changed messages to determine if you need to make any updates.

Db2 Admin Tool condition codes

Db2 Admin Tool Reverse Engineering jobs and Bind Manager jobs return condition codes to indicate whether the job was successful.

Db2 Admin Tool Reverse Engineering condition codes

A Db2 Admin Tool Reverse Engineering job returns one of the following condition codes:

0

The job completed successfully.

4

The job returned a warning condition.

For example, the following situations result in warning conditions:

- A parameter error occurred. In this case, either the parameter is ignored or the default is used. No generate requests are issued.
- The requested object is not found.

8

An error occurred, and processing ends. Other errors might be issued.

For example, the following situations result in an error with condition code 8:

- No parameters are found.
- The Db2 version is not yet supported.

12

An internal error or limitation occurred, and other severe errors are detected. Processing ends.

For example, the following situations result in an error with condition code 12:

- The Db2 version is not supported.
- The remote location is not defined or is not a Db2 for z/OS system.

16

A severe error occurred.

Db2 Bind Avoidance condition codes

A Db2 Admin Tool Bind Avoidance job returns one of the following condition codes:

0

The job completed successfully. The new source has passed SQL validation checks, and a Db2 bind is not required.

4

Either the SQL validation failed, or an error occurred. A Db2 bind is required. Check the associated diagnostic messages for details.

8

A possible precompiler error occurred. Check the associated messages for a description of the specific error.

12

An error occurred. Check the associated messages for a description of the specific error.

16

A severe error occurred. Bind Manager cannot make a determination about whether a bind is required. Check the associated messages for a description of the specific error.

Db2 Admin Tool messages

Db2 Admin Tool messages begin with the prefix ADB. The exception is any messages that are specific to the Bind Manager function; those messages begin with BND.

Not all Db2 Admin Tool messages are included in this section.

ADB100E **A parameter was omitted or an invalid parameter value was passed to module name *module_name*.**

Explanation:
A required parameter has not been located in the parameter string passed to the program.

System action:
None.

User response:
If the parameter string was edited, provide the missing parameter, and ensure the parameter value is valid. If the parameter was omitted by the product, contact IBM Software Support.

ADB200E ***message-string***

Explanation

message-string
Error message text. This text varies depending on the context, as it is set by various panels and programs.

User response:
Follow the instructions in the message text to resolve any errors.

ADB200W ***message string***

Explanation

message-string
Informational or warning message text. This text varies depending on the context, as it is set by various panels and programs.

User response:
Follow the instructions in the message text.

ADB207I **This view will be dropped - no revoke privilege info to show**

Explanation:
The I line command was entered against a view. However, no interpretation information is available for the view, because it will be dropped due to dependent objects having privileges revoked. For example, in the case of a view (vw2) on a view (vw1), when privileges are revoked on vw1, vw2 needs to be dropped.

System action:
Processing continues.

User response

No action is required.

ADB226E **DB2 commands not available**

Explanation:
The currently connected DB2 system does not support DB2 commands.

System action:
Processing stops.

User response:
Ensure that you are connected to a system that is running DB2 for z/OS and that it accepts DB2 commands.

ADB228E **Invalid table name**

Explanation:
The table name *table_name* is not allowed.

System action:
Processing stops.

User response:

Specify a valid table name and try the operation again.

ADB229E Panel error

Explanation:
An ISPF error occurred on display of panel *panel_name*, RC=*return_code*.

System action:
Processing stops.

User response:
Ensure that the specified panel is correct. If you are using the PANEL command ensure that the specified panel name is correct and that the panel can be used in this context. If the problem persists then contact IBM Software Support.

ADB230S No table displayed

Explanation:
The *command* command requires an active table to act on.

System action:
Processing stops.

User response:
Specify a valid table for the command and try the operation again.

ADB231E No table specified

Explanation:
The *command* command did not specify a table name.

System action:
Processing stops.

User response:
Specify a table name and try the operation again.

ADB232E Table error

Explanation:
An error occurred while processing ISPF table: *table_name*.

System action:
Processing stops.

User response:
Ensure that the table is a valid ISPF table.

ADB233E Invalid sort field

Explanation:
The specified sort field *field_name* is not present in the table.

System action:
Processing stops.

User response:
Specify a sort field that is in the table or use the sort command without parameters to display the fields that are in the table.

ADB249E Invalid data set

Explanation:
A command list data set must have DSORG=PO, RECFM=FB, and LRECL=80. The LISTDSI return code=*return_code*, and reason code=*reason_code*. LISTDSI reports that the data set has DSORG=*SYSDSORG*, RECFM=*SYSRECFM*, LRECL=*SYSLRECL*.'

System action:
Processing stops.

User response:
Specify a valid data set with the required attributes.

ADB252S Program Error

Explanation:
The display driver is stopped due to an internal error, oncode=*oncode*.

System action:
Processing stops.

User response:
Try the operation again. If the problem persists, contact IBM Software Support.

ADB255E DB2 IFI error, return code=*rc*, reason code=*reason*, command=*db2_command*

Explanation

The specified Db2 command returned an error.

rc
The return code from the command.

reason
The reason code from the command.

db2_command
The Db2 command. If the command is longer than 80 characters, only the first 80 characters are displayed in the message.

System action:
Processing stops.

User response:
Look up the reason code in the Db2 documentation and take any necessary actions to correct the error.

Related information

Db2 reason codes (Db2 12 for z/OS documentation)

ADB255W DB2 IFI warning, return code=*rc*, reason code=*reason*, command=*db2_command*

Explanation

The specified Db2 command returned a warning.

rc
The return code from the command.

reason
The reason code from the command.

db2_command
The Db2 command. If the command is longer than 80 characters, only the first 80 characters are displayed in the message.

System action:
Processing continues.

User response:
Look up the reason code in the Db2 documentation to determine if any action is needed.

Related information

Db2 reason codes (Db2 12 for z/OS documentation)

ADB259S DB2 V*version* unsupported

Explanation:
The version of DB2 that you are using is not supported by the version of DB2 Admin that you are using.

System action:
Processing stops.

User response:
Ensure that the version of DB2 Admin that you are using supports the version of DB2 that you are using.

ADB260I *n* row(s) affected by the *operation* statement

Explanation

The specified number of rows were inserted, updated, or deleted.

n
The number of rows affected.

operation
The operation that was executed. Possible values are INSERT, UPDATE, or DELETE.

System action:
Processing continues.

User response

No action is required.

ADB266I User command list refreshed

Explanation:
Any updates to the user-defined primary command lists were saved.

Explanation:
Processing continues.

User response:
You can now issue the updated commands as needed.

Related tasks

“Defining your own primary commands” on page 1039
When you define a primary command in Db2 Admin Tool, it can then be issued from any command line in the product.

ADB267I *Operation* was successful.

Explanation

The *Operation* was successful. The SQL statement that was performed was a DB2 MERGE statement, so the target row was either updated or a new row was inserted.

Operation performed:

Update
An existing row was located and updated.

Insert
An existing row was not located but a new row was inserted.

Explanation:
Processing continues.

User response:
None.

ADB268I *Operation* was successful. The originally specified row was not updated.

Explanation

The *Operation* was successful. The SQL statement that was performed was a DB2 MERGE statement. The product detected that the user originated an action from one entry but changed the value used by DB2 MERGE to locate the row. This might lead to a new row being inserted or a different row being updated than was originally intended.

Operation performed:

Update
An existing row was located and updated.

Insert
An existing row was not located but a new row was inserted.

Explanation:
Processing continues.

User response:
None.

ADB26AE Not a result set

Explanation

This line command is valid only for result sets. Result sets are indicated by a value of **<RESULT>** in the **Type** column.

System action:
Processing stops.

User response:
Specify this line command for a result set.

ADB272E Not possible - The operation requested is not possible on this panel for this object.

Explanation:
The requested operation cannot be completed.

User response:
Select a valid option for the object or navigate to a different panel where the operation is allowed.

ADB279E Mutually exclusive - STOGROUP, PRIQTY, SECQTY or ERASE cannot be combined with USING VCAT or applied to user-managed objects.

Explanation:
An invalid combination was specified.

System action:
Processing stops.

User response:
Specify valid values.

ADB279F Altering VCAT and STOGROUP at the same time is not permitted. Choose one to alter.

Explanation:
Both VCAT and STOGROUP alterations were specified.

System action:
Processing stops.

User response:
Alter VCAT and STOGROUP in separate operations.

ADB279G Altering VCAT is not permitted for Partitioned-by-Growth table space.

Explanation:
A VCAT alteration was specified for a partition-by-growth (PBG) table space.

System action:
Processing stops.

User response:
Specify a different alteration.

ADB279H Altering Piece Size and Cluster at the same time is not permitted. Alter one at a time.

Explanation:
Both piece size and cluster alterations were specified.

System action:
Processing stops.

User response:
Alter piece size and cluster in separate operations.

ADB279I Backup process complete. Confirm that backup DDL has been created properly.

Explanation:
The backup was created.

System action:
Processing continues.

User response:
Check the DDL backup.

ADB279J Creation of Run job complete. Inspect and submit to process.

Explanation:
The run job was created.

System action:
Processing continues.

User response:
Submit the run job.

ADB279K Warning: Backup DDL process has been skipped since scan of WSL has detected no SQL DROP statements.

Explanation:
No backup was created, because the work statement list (WSL) does not include any SQL DROP statements.

System action:
Processing continues.

User response

No action is required.

ADB279L Warning: Problem discovering Db2 SSIDs. List might be incomplete.

Explanation:
Db2 Admin Tool tried to determine all active Db2 subsystems. However, due to external factors, the list of subsystems might be incomplete.

System action:
Processing continues.

User response:
Retry the operation.

ADB294E The unloading of LOB columns can require the use of templates. The

templates can be generated only through work statement list (WSL) processing. Add the statements to a WSL and then run the WSL in batch mode to accomplish this task.

Explanation:
LOB columns must be unloaded to perform the operation. The unloading of LOB columns requires the use of templates, and templates can be generated only through work statement list (WSL) processing.

System action:
Processing stops.

User response:
Specify that the statements are to be added to a work statement list (WSL). When the WSL is run, ensure that it is run in batch mode.

ADB298E **RESET MAXASSIGNED cannot be YES if the RESTART WITH value is changed.**

Explanation:
RESET MAXASSIGNED must be NO if the RESTART WITH value is changed.

System action:
Processing stops.

User response:
Set RESET MAXASSIGNED to NO if you change RESTART WITH.

ADB300E **Module *module_name* DD statement is missing.**

Explanation:
The specified DD statement is missing.

System action:
Processing stops.

User response:
Supply the missing DD statement and try again. Alternatively, regenerate the job and try again.

ADB318E **The value must be *value_1*, or an integer between *value_2* and *value_3*.**

Explanation:
The specified value is not allowed. The value must be equal to *value_1* or an integer between *value_2* and *value_3*.

System action:
Processing stops.

User response:
Enter a valid value and try the operation again.

ADB325E **The specified line command is invalid for the object type *object_type*.**

Explanation

The requested operation is not valid for the object.

object_type
The type of object, as listed in the **T** (type) column.

System action:
Processing stops.

User response:
Either issue a valid line command for the object type, or issue this line command against another object for which this operation is valid.

ADB330U **The sum of the values for PCTFREE (PF) and FOR UPDATE (PFU) must be less than or equal to 99.**

Explanation:
This restriction on the PCTFREE and FOR UPDATE values is Db2 restriction.

System action:
Processing stops.

User response:
Correct the PF and PFU values.

ADB330V **INSERT ALGORITHM cannot be altered to value of zero. Only values allowed are 1 or 2.**

Explanation:
After the INSERT ALGORITHM value for a table space has been changed to a non-zero value, you cannot change it back to zero.

System action:
Processing stops.

User response:
Specify a value of 1 or 2 for INSERT ALGORITHM.

ADB332I **The string &`db2aetok` was found.**

Explanation:
The requested string was found in the information displayed.

System action:
Processing ends.

User response:
None.

ADB332W **The string &`db2aetok` was not found.**

Explanation:
The requested string was not found in the information displayed.

System action:
Processing ends.

User response:
None.

ADB338E **Invalid specification. Changing procedure types during CREATE is not allowed from this panel.**

Explanation

The following changes are not allowed when using the CRE (Create Like) line command from panel ADB21O:

- Changing from an external procedure, such as PLI, to an SQL procedure
- Changing from an SQL procedure to a non-SQL procedure
- Changing from an SQL external procedure to a native stored procedure

Changing from an external procedure to another of a different language is allowed, but not recommended.

System action:
None.

User response:
Restore the original language or native stored procedure value to the appropriate field.

ADB343A **Invalid specification - *option_1* and *option_2* are mutually exclusive options.**

Explanation:
The specified utility options (*option_1* and *option_2*) are not compatible.

System action:
Processing stops.

User response:
Check the Db2 for z/OS documentation for the syntax for the utility that you want to run. Then, correct the utility options in Db2 Admin Tool.

Related information

Db2 online utilities (Db2 12 for z/OS)

ADB343W **The value of DISCARDS is not compatible with BACKOUT YES. Specify a value for DISCARDS that is greater than 0.**

Explanation:
If you specify YES for the BACKOUT option, the value of the DISCARDS option must be greater than 0.

System action:
Processing stops.

User response:
Specify a value for DISCARDS that is greater than 0.

Related reference

Syntax and options of the LOAD control statement (Db2 12 for z/OS)

ADB343E **The specified keyword *keyword* can not be specified because *reason*.**

Explanation

The specified keyword is not valid because of the reasons listed below. If keyword is PARALLEL for the LOAD utility, there are two reasons:

1. The table to be loaded has LOB or XML columns and SHRLEVEL NONE is specified.
2. The table to be loaded has XML columns and is in a simple or segmented table space and SHRLEVEL CHANGE is specified.

System action:
Processing stops.

User response:
If keyword is PARALLEL, specify a valid keyword and try the operation again.

ADB346A ***temporary override* will be used in the REORG TABLESPACE utility statement instead of *customized option setting* to ensure pending definition changes for MOVE TABLE are materialized.**

Explanation

To materialize a pending definition change, the REORG TABLESPACE utility statement must include SHRLEVEL REFERENCE or CHANGE and must not include FASTSWITCH NO. Because the specified REORG options do not satisfy this requirement, Db2 Admin Tool replaced the options as indicated in the message text to ensure that the MOVE TABLE changes are materialized.

temporary override
The REORG option that will be used. Possible values are:

- `SHRLEVEL CHANGE`
- `FASTSWITCH YES`
- `SHRLEVEL CHANGE and FASTSWITCH YES`

customized option setting
The REORG option that was specified on the **Specify Utility Options - REORG (ADB2USO)** panel. Possible values are:

- `SHRLEVEL blank/NONE`
- `FASTSWITCH NO`

- `SHRLEVEL blank/NONE and FASTSWITCH NO`

System action:
Processing continues.

User response

No action is required.

ADB359E **The LC line command cannot be used on a view that is defined on more than one table. Use the T line command to locate the table that you want to process.**

Explanation:
The view selected is defined on more than one table but the LC line command specified can only operate on a single table. The line command cannot be executed because the target of the load is ambiguous.

System action:
Processing stops.

User response:
Use the T line command to display the tables associated with the view. Locate the table that you want to process and then issue the U.LC line command against the specific table.

ADB362E **Enter string**

Explanation:
A character string was not specified in the FIND command.

System action:
Processing stops.

User response:
Enter the string of characters to be found.

ADB363E **Invalid string**

Explanation:
The FIND string cannot be a null ('''') string.

System action:
Processing stops.

User response:
Specify a non-null string of characters to search for in the FIND command.

ADB364E **Invalid column number**

Explanation:
The column number in the FIND command is invalid.

System action:
Processing stops.

User response:
Specify a valid column number and issue the FIND command again.

ADB365E **FROM column > TO column**

Explanation:
The FROM column that was specified in a FIND command is greater than the TO column that was specified in the command.

System action:
Processing stops.

User response:
Specify a FROM column number that is less than the TO column number and issue the FIND command again.

ADB366E **Enter a FIND command**

Explanation:
The RFIND command is used to reissue the FIND command that was previously issued.

System action:
Processing stops.

User response:
Issue a FIND command first then issue the RFIND command.

ADB372E **The table is not in a UTS.**

Explanation:
Inline length is only allowed for tables within a Universal Table Space (UTS). An inline length has been specified for a table that is not within a UTS.

System action:
Processing stops.

User response:
Remove the inline length specification for this table.

ADB373E **Inline length cannot be greater than Data length.**

Explanation:
The Inline length value must be less than or equal to the Data length value.

System action:
Processing stops.

User response:
Correct the inline length value.

ADB376E **Inline length cannot be less than the length of the default column value.**

Explanation:
The inline length value must be greater than or equal to the length of the value for the column default.

System action:
Processing stops.

User response:
Increase the inline length value.

ADB377E The parameter is too large. The total number of partitions exceeds the MAXPARTITIONS limit of *<parml>*.

Explanation:
The number of partitions specified on the ADDPART command will result in a total partition number which exceeds the MAXPARTITIONS value for this table space.

System action:
Processing stops.

User response:
Specify a lower value for the ADDPART option.

ADB378E There is an option conflict. A field procedure cannot be specified with a *<parml>* data type.

Explanation:
Specification of a field procedure is not allowed with this data type.

System action:
Processing stops.

User response:
Either change the data type, or do not specify a field procedure name.

ADB379E A SECLABEL is not allowed for tables enforced by row access control.

Explanation:
A security label is not allowed for tables with activated row-level access control.

System action:
Processing stops.

User response:
None.

ADB398E The encoding scheme of the specified table space must be *V_CCSID*.

Explanation:
The encoding scheme of the EXPLAIN table must be the same as the table space which contains the EXPLAIN table. In DB2 9 new-function mode and previous releases and modes (for example, DB2 9 enabling-new-function mode, or DB2 9 compatibility mode), because the encoding scheme of the EXPLAIN table must be EBCDIC or UNICODE, the encoding scheme of the specified table space which contains the EXPLAIN table must be EBCDIC or UNICODE. In Db2 10 conversion mode and more current releases, and in modes that follow conversion mode, because the encoding scheme of the EXPLAIN table must be UNICODE, the encoding scheme of the specified table space which contains the EXPLAIN table must be UNICODE.

System action:
None.

User response:
In DB2 9 new-function mode or previous releases and earlier modes, specify a table space which is encoded in EBCDIC or UNICODE. In Db2 10 conversion mode, and more current releases and modes that follow conversion mode, specify a table space which is encoded in UNICODE.

ADB397W *table-name* is a created temporary table. Only ALL or ALL PRIVILEGES can be granted to a created temporary table.

Explanation:
The GRANT command operates on the entire list of tables that is on the Tables, Views, and Aliases panel. When different types of tables are listed, the GRANT command will fail, if any known restriction applies to any of the tables.

System action:
Processing continues.

User response:
Issue the GR line command for each table. Alternatively, you can use a different table filter on the Tables, Views, and Aliases panel so that only created temporary tables are listed.

ADB399W This action may lead to an error when you apply changes later because the altered table, *table_name*, requires the table space that is created by the altered table space, *tablespace_name*.

Explanation:
When you alter a table space (ALT TS) by changing the DBname or TSname and if the alter table (ALT TB) statement specified the same DBname or TSname, the Admin Tool checks the catalog before invoking the CREATE TS statement. The Administration Tool checks the previous ALT TS action to determine whether the same table space will be created. If yes, the CREATE TS statement at TB level is ignored and the altered table requires the table space that is created by the altered table space. When you use an A or D line command on the altered TB or altered TS which has a dependency relationship, the table space needed by the altered table might not be created, which can lead to an error when you apply changes later.

System action:
Processing continues.

User response:

No action is required.

ADB448W **Limit key values are not in proper ascending or descending order.**

Explanation:
The new limit key values were not inserted in the correct numeric sequence.

System action:
Processing stops.

User response:
Correct the limit key values.

ADB456E **The database already exists. Enter a new database name.**

Explanation:
The database cannot be renamed to an existing database name.

System action:
None.

User response:
Enter a new database name in the **New database name** field and press Enter.

ADB461E **A system-managed table must have columns defined as ROW BEGIN and ROW END for the PERIOD clause. Either one or both columns of this type are missing in this table.**

Explanation:
A request for a System period has been made without valid columns for the start and end columns of the period in the table definition.

System action:
None.

User response:
Return to the column definition panel and assure that there are columns with the ROW BEGIN and ROW END attributes defined before proceeding.

ADB462E **Specify both a start and an end column.**

Explanation:
You must specify both a start and end column for the BUSINESS_TIME period on the Select BUSINESS TIME Period Columns panel .

System action:
None.

User response:
Use the S and E line commands to select the Start and End columns for the BUSINESS_TIME period. Use CANCEL to return to the Create Table Columns panel without making a selection. .

ADB463E **Only one start and one end column are allowed.**

Explanation:
You cannot specify more than one start and one end column for the BUSINESS period.

System action:
None.

User response:
Use the R command to remove any duplicate selection.

ADB464E **You must have at least two TIMESTAMP(6) WITHOUT TIME ZONE or two DATE columns valid for BUSINESS_TIME period columns before proceeding.**

Explanation:
There must be at least two columns which are valid for the business period start and end columns before proceeding to the Select BUSINESS TIME Period Columns panel.

System action:
None.

User response:
Add or redefine columns on the Create Table Columns panel to assure that there are two columns valid for the business period.

ADB465E **A request for BUSINESS_TIME WITHOUT OVERLAPS for the constraint without a BUSINESS_TIME period will be ignored.**

Explanation:
This request will be ignored if you specify YES for the BUSINESS_TIME WITHOUT OVERLAPS option when defining a primary key, if you have not already defined a BUSINESS_TIME period. If you do not define a BUSINESS_TIME period before issuing the CREATE command, the option will be ignored.

System action:
None.

User response:
No action is required. If you want to use the option, define a BUSINESS_TIME period.

ADB466E **The BUSINESS_TIME WITHOUT OVERLAPS option is invalid because a BUSINESS_TIME period start or end column matches a column in the primary key.**

Explanation:

The BUSINESS_TIME WITHOUT OVERLAPS option is not valid if a start or end column of the business period matches any of the keys of the primary constraint.

System action:
None.

User response:
Either change the business period start or end column, or change the primary key columns so that they do not conflict.

ADB467E **There might be some options from the model table which are not used.**

Explanation:
MODEL=YES was specified from the main Create Table panel. Certain options might not be adopted from the model table.

System action:
None.

User response:
No action is required.

ADB468E **Use the TBLOPTS command to specify a SYSTEM_TIME period.**

Explanation:
Columns with attributes AS ROW BEGIN and AS ROW END have been specified.

System action:
None.

User response:
Go to the Create Table Options panel to specify a SYSTEM_TIME period. .

ADB471E **The specified database name is implicit. Enter a new database name.**

Explanation:
The database cannot be renamed to an implicit database name, such as DSN*nnnnn* where *nnnnn* is a numeric value.

System action:
None.

User response:
Enter a new database name in the **New database name** field and press Enter.

ADB472W **No utilities (except UNLOAD) will be generated for implicit table spaces.**

Explanation:
The RENDB function will not generate utilities for implicit table spaces.

System action:
This warning message is displayed if the database to be renamed has at least one implicit table space.

User response:
Press Enter if you want to continue processing.

ADB473E **The specified database name is reserved. Enter a new database name.**

Explanation:
The database cannot be renamed to a reserved database name of DSNDB01, DSNDB04, DSNDB06, or DSNDB07.

System action:
None.

User response:
Enter a new database name in the **New database name** field and press Enter.

ADB526E **An XML column defined as NOT NULL and no default cannot be added.**

Explanation:
An XML column cannot be added with the NOT NULL attribute and no default, since there is no default data value to LOAD for columns.

System action:
Processing stops.

User response:
Re-specify the attributes to allow null values.

ADB539E **The target SSID *DB2_SSID* cannot be found in customization table. Ensure that the SSID customization table is properly defined.**

Explanation:
The SSID for the target Db2 subsystem cannot be found.

System action:
Processing stops.

User response:
Ensure that the SSID is defined in the ADBTPARM member. Using Tools Customizer, edit the SSID, generate the customization jobs, and submit the ADBCUST job that corresponds to the SSID that you edited. When the ADBCUST job is submitted, the SSID will be added to the ADBTPARM member.

ADB559F **The LOAD job member names to be generated exceed eight characters. Specify a prefix that is less than five characters for the job member names.**

Explanation

Because numerous tables are being processed, the LOAD job member names to be generated exceed eight characters (ADBTSnnRL) This error occurs when the table space being Altered or Redefined has more than nine tables and the following options are specified:

Combine job steps=NO
Member name or prefix=ADBTS (five chars)
Unload Method=H

System action:
Processing stops.

User response:
Specify a prefix that is less than five characters for the job member names.

ADB559G **YES is not allowed when moving to a Partitioned-by-Growth table space.**

Explanation:
It is not permitted to redefine a table space to Partitioned-by-Growth or Partitioned-by-Range with the Member Cluster input field set to YES. This restriction is a DB2 9 restriction.

System action:
Processing stops.

User response:
Specify NO in the Member Cluster input field when you redefine a Table Space to Partitioned-by-Growth or Partitioned-by-Range.

ADB559O ***creator.name* contains *n* tables. Converting to a partitioned table space is not supported, therefore options for partitions cannot be changed.**

Explanation:
The database cannot be partitioned because it contains more than one table.

System action:
Processing stops.

User response:
You can continue with other line commands or press PF3 to leave the panel.

ADB559P **Only converting to a Partition-by-Growth (PBG) or a Partition-by-Range (PBR) table space is permitted.**

Explanation:
The table space is PBG or PBR, but the number of partitions or the segment size cannot be changed.

System action:
Processing stops.

User response:
Enter ORIGINAL on the command line to reset the values to the original values.

ADB559Q **A table space name is required when moving to *type*.**

Explanation

The number of partitions or the segment size was changed, but no table space name was provided. The *type* can be:

- Partitioned-by-Growth table space (PBG)
- Partitioned-by-Range table space (PBR)
- Partitioned table space

System action:
Processing stops.

User response:
Provide a table space name, or type ORIGINAL on the command line to reset the values to the original values.

ADB586E **An UNLDDN template is required when SPANNED=YES is set.**

Explanation:
An UNLDDN template must be provided in order for DB2 to determine the space necessary and to create an unload file with the spanned attribute.

System action:
Processing stops.

User response

On panel ADB28M, specify the TU option. On panel ADB25TU3, specify UTILITY TEMPLATE Usage, and define a template for UNLDDN. Return to panel ADB28M to set the 'Generate template statements' field to YES, then press Enter.

Alternatively, change the 'Spanned' field to NO.

ADB587E **The HIDDEN attribute is not allowed for a column defined as ROWID.**

Explanation:
If a column is defined with a ROWID data type, then the column cannot be specified as HIDDEN.

System action:
Processing stops.

User response:
Either change the HIDDEN attribute to NO, or specify a different column type.

ADB588E **You must change one or more keys in order to change the primary key constraint name.**

Explanation:
You cannot change the primary key constraint name without also changing one or more keys for the constraint.

System action:
Processing stops.

User response:
Change one or more of the constraint columns, or restore the original constraint name.

ADB589E **You must choose one or more columns for the constraint key.**

Explanation:
You must specify one or more columns for the constraint key when adding a primary or unique key constraint.

System action:
Processing stops.

User response:
Specify one or more columns for the constraint key before proceeding.

ADB600E **Invalid time. The specified value must be formatted as (+/-)hh:mm. The hh parameter must be between -12 and +14 and mm between 00 and 59.**

Explanation:
The time value is not specified in the correct format. The value must be formatted as (+/-)hh:mm. The hh parameter must be a numeric value between -12 and +14 and the mm parameter must be a number between 00 and 59.

System action:
Processing stops.

User response:
Specify the time value using valid formatting and try the operation again.

ADB614I **The Real-Time Statistics for the object have been refreshed.**

Explanation:
The REFRTS command completed successfully and the real-time statistics have been updated.

System action:
Processing continues.

User response:
No action is required.

ADB614E **The CCSID values must match.**

Explanation:
The encoding scheme specified for the array subtype must match the encoding scheme for the source type.

System action:
Processing ends.

User response:
Change the encoding scheme inputs so that they match and try the operation again.

ADB615E **Invalid length value.**

Explanation:
Do not specify length with *array* array subtype. Length can be specified only for VARCHAR array subtype.

System action:
Processing ends.

User response:
Remove the length value if you are using an INTEGER array subtype and try the operation again.

ADB616E **Invalid CCSID option.**

Explanation:
array is a valid array subtype. CCSID can be specified only for VARCHAR array subtype.

System action:
Processing ends.

User response:
Remove the CCSID value input or change the array subtype and try the operation again.

ADB617E **Invalid data subtype.**

Explanation:
type data subtype is invalid with *array* array subtype.

System action:
Processing ends.

User response:
Change the data subtype and array subtype to values that are valid and try the operation again.

ADB618E **Invalid data type.**

Explanation:
The user data type specified is invalid for the CREATE *procedure/function* statement that is being built.

System action:
Processing ends.

User response:
Change the data type to a value that is valid and try the operation again.

ADB700E **Column not allowed. Column *column_name* cannot be specified as part of the primary key because it is a DECFLOAT, XML, or LOB**

data type, or it is a row-change-timestamp column.

Explanation:
DB2 does not allow a column of the indicated type to be included as one of the primary key columns for the table.

System action:
Processing stops.

User response:
Remove the column from the primary key specification.

ADB701E Column not allowed. Column *column_name* cannot be specified as part of the unique key because it is a LOB data type.

Explanation:
DB2 does not allow a unique key to be created for a LOB column data type.

System action:
Processing stops.

User response:
Remove the column from the specification.

ADB702E Column not allowed. Column *column_name* cannot be specified as part of the foreign key because it is a LOB data type.

Explanation:
DB2 does not allow a column of the indicated type to be included in a foreign key definition.

System action:
Processing stops.

User response:
Remove the column from the specification.

ADB703E Column not allowed. Column *column_name* cannot be specified as a column of a parent key in a REFERENCES clause because it is a LOB data type.

Explanation:
DB2 does not allow a column of the indicated type to be included in a REFERENCES clause.

System action:
Processing stops.

User response:
Remove the column from the specification.

ADB704E Column not allowed. Column *column_name* cannot be specified as a column of a partitioning key because it is a *data_type* data type.

Explanation:
DB2 does not allow a column of the indicated type to be included as one of the partitioning columns for the table.

System action:
Processing stops.

User response:
Remove the column from the specification.

ADB705E Operation not allowed. Column *column_name* cannot be changed to a LOB column because a check constraint exists on this column.

Explanation:
DB2 does not allow a column of the indicated type to be included in a check constraint.

System action:
Processing stops.

User response:
Remove the column from the specification.

ADB706E Operation not allowed. Column *column_name* cannot be changed to a LOB column because a field procedure exists on this column.

Explanation:
A column with a field procedure cannot be changed to a LOB data type.

System action:
Processing stops.

User response:
Remove the field procedure prior to changing the column definition.

ADB707E Operation not allowed. Column *column_name* cannot have a default value. Only NULL is allowed.

Explanation:
DB2 does not allow the column to have a default value. Specifying NULL is allowed.

System action:
Processing stops.

User response:
Specify NULL as required.

ADB708E Operation not allowed. Column *column_name* cannot be converted from a LOB data type to any other data type.

Explanation:
Data type conversion from a LOB data type is not allowed.

System action:
Processing stops.

User response:
Specify a data type conversion that is allowed.

ADB709E Column not allowed. Column *column_name* cannot be provided as a column in the constraint because it is a DECFLOAT, XML or LOB data type, or it is a row-change- timestamp column.

Explanation:
DB2 does not allow a column of the indicated type to be included in the constraint

System action:
Processing stops.

User response:
Remove the column from the specification.

ADB710E Operation not allowed. Column *column_name* cannot be converted from NULL to NOT NULL.

Explanation:
The column cannot be converted from NULL to NOT NULL.

System action:
Processing stops.

User response:
Retain the NULL specification.

ADB711E This operation is not allowed against a hidden column.

Explanation:
The line command that you entered is not allowed on a hidden column.

System action:
Processing stops.

User response:
Do not issue the command against the column.

ADB712E Improper length. A LOB column cannot be shortened in length.

Explanation:
A LOB column's length cannot be reduced.

System action:
Processing stops.

User response:
Retain the original column's length.

ADB720E Column *column_name* cannot be specified as a column of an index key due to its data type, *data_type*.

Explanation:
A column of the selected data type cannot be specified as part of an index.

System action:
Processing stops.

User response:
Select a column with a data type that can be part of an index.

ADB722E *value_1* is not allowed with *value_2*.

Explanation

This combination of values is not allowed.

value_1
An option, specification, or condition that is not allowed with *value_2*.

value_2
An option, specification, or condition that is not allowed with *value_1*.

System action:
Processing stops.

User response

Correct the specification.

For example, if the message text is The `RECLUSTER option is not allowed with option SORTDATA specified as YES or BLANK`, specify SORTDATA NO with the RECLUSTER option.

ADB723E Operation not allowed. A table defined with DATA CAPTURE CHANGES cannot be placed into a NOT LOGGED table space.

Explanation:
A table defined with the DATA CAPTURE CHANGES attribute cannot be placed into a table space defined with the NOT LOGGED attribute.

System action:
Processing stops.

User response:
Specify a table space with the proper DB2 logging attribute, or remove the DATA CAPTURE CHANGES attribute from the table.

ADB724E Operation not allowed. A table cannot be moved to an implicitly created database or table space.

Explanation:
A table cannot be placed into a table space which was implicitly created by DB2. .

System action:
Processing stops.

User response:
Specify a table space that was explicitly created.

ADB725E A row change timestamp column cannot be added to the table.

Explanation:
Adding a ROW CHANGE TIMESTAMP column is not permitted.

System action:
Processing stops.

User response:
Respecify the column without the ROW CHANGE TIMESTAMP attribute.

ADB726E Conversion to or from a row change timestamp column is not allowed.

Explanation:
Changing to or from a ROW CHANGE TIMESTAMP column is not permitted

System action:
Processing stops.

User response:
Respecify the column without changing to or from a ROW CHANGE TIMESTAMP attribute.

ADB727W Different columns in the primary key definition were specified, or the length of a primary key column was increased.

Explanation:
The primary key columns for the table were respecified, or a column length was changed. The primary key must be dropped first, which will result in the loss of any referential integrity definition based upon the primary key definition.

System action:
This message is issued as a warning and an action prompt panel is displayed.

User response:
An ALTER TABLE ... DROP PRIMARY KEY statement is needed to perform this change. Use option 3 to create new referential constraints and any required new indexes.

ADB728E Conversion from *column_type* to *new_column_type* data type, or changing the length of a *column_type* data type is not allowed.

Explanation:
Changing the data type to or from the indicated data type is not permitted.

System action:
Processing stops.

User response:
Respecify the column without changing the data type.

ADB729E Conversion from *column_type* to *new_column_type* data type is not allowed since the source column is not defined as FOR BIT DATA.

Explanation:
The original column is not defined as FOR BIT DATA. Conversion is only allowed on FOR BIT DATA columns.

System action:
Processing stops.

User response:
None.

ADB730E Operation not allowed. The target table space must be of the same partitioning type as the existing table space (partition by growth or partition by range).

Explanation:
Moving a table to a table space of a different format when the old or new table space is partitioned by growth is not permitted.

System action:
Processing stops.

User response:
Specify a target table space of the same type as the table's current table space.

ADB731E Too many operations performed. Only one operation is allowed at a time.

Explanation:
The combination of operations is not allowed.

System action:
Processing stops.

User response:
Specify one operation at a time.

ADB731E Too many operations performed. Only one operation is allowed at a time.

Explanation:
The combination of operations is not allowed.

System action:
Processing stops.

User response:
Specify one operation at a time.

ADB735E An upgrade cannot be done. The table *table_name* can only be upgraded from the previous

release to the current release. Re-create the table.

Explanation:
An upgrade cannot be done to the control table *table_name* because it is not at the proper level.

System action:
None.

User response:
Drop and re-create the TEMPLATE control table.

ADB737E **Incorrect table format. The table *table_name* does not have the expected column names, data types, or both. Check the current definition of the TEMPLATE control table.**

Explanation:
The identified TEMPLATE control table cannot be upgraded because the table definition is incorrect.

System action:
None.

User response:
Check the table name and the table owner to see if it is a control table. LISTDEF and TEMPLATE control tables are DB2 control tables. Thus, they could be created during DB2 installation by the DSNTIJCC member. DB2 Administration Tool could also be used to create LISTDEF and TEMPLATE control tables. The default name for LISTDEF control tables is DSNACC.UTLIST, and the default name for TEMPLATE control tables is DSNACC.UTTEMPLATE. See LISTDEFs and TEMPLATEs in this User Guide for further information.

ADB748E **There has been an unsupported request *request_type* for exec ADBEUSV.**

Explanation:
A mismatch might exist between panel ADB2USV and exec ADBEUSV.

System action:
Processing stops.

User response

Log off, log on, and try the procedure again.

If the problem persists, contact IBM Software Support.

ADB79AW **A unique key constraint named, *constraint_name* already exists for this table. You can change the constraint name, enter CONTINUE to replace the keys for the constraint, or END to exit.**

Explanation:
The table already has a unique key constraint with this name, or a constraint was added within this ALT session.

System action:
This is a warning message.

User response

Try these actions to correct the problem:

- Change the constraint name
- Enter CONTINUE to replace the key, or END to exit without saving.

ADB799E **The table space is not a range partitioned table space.**

Explanation:
The LKEY line command was issued, but it is not valid for partition by growth table spaces. This line command is only valid for range partitioned table spaces.

System action:
The system waits for the next command.

User response:
Issue a different command and press Enter or press PF3 to leave the panel.

ADB799W **A primary key constraint already exists for this table. Enter CONTINUE to replace the key, or END to exit.**

Explanation:
The table already has a primary key, or one was added within this ALT session. Only one primary key is allowed.

System action:
This is a warning message.

User response:
Enter CONTINUE to replace the key, or END to exit without saving.

ADB811E **NO is not valid for this option because Drop Impact Report is specified as YES or BATCH.**

Explanation:
If you set the **Show this panel prior to each drop** field to NO, then you cannot set the **Display Drop Impact Report** field to YES or BATCH. The settings are not compatible.

System action:
Processing stops.

User response:
Set both the **Display Drop Impact Report** field and **Show this panel prior to each drop** field to NO. Alternatively, specify YES or blank in the **Show this**

panel prior to each drop field and specify YES in the **Display Drop Impact Report** field.

ADB812E **Lines that are marked with '?' are not committed to change. Remove the '?' and press Enter to commit the change.**

Explanation:
A value for an object was marked to change but the change is not committed by pressing Enter once. You must press Enter again.

System action:
DB2 Admin puts a question mark in the line command field and puts the statement 'modify pending' in the message column.

User response:
Remove the question mark from the lines that you want modified and press Enter to continue.

ADB813E **RESET MAXASSIGNED cannot be YES if the RESTART WITH value is changed.**

Explanation:
RESET MAXASSIGNED must be NO if the RESTART WITH value is changed.

System action:
Processing stops.

User response:
Set RESET MAXASSIGNED to NO if you change RESTART WITH.

ADB815E **This table is not an archive enabled table.**

Explanation:
The ARCH line command was issued for a table that is not archive enabled. The command cannot be processed.

System action:
The system waits for the next user action.

User response:
Issue the ARCH line command for table objects that have been archive enabled. Use the BROWSE primary command from panel ADB21T to see catalog information from SYSTABLES. Archive enabled tables are those with TYPE=T and with the ARCHIVING_SCHEMA ARCHIVING_TABLE columns having the schema and name of the archive table.

ADB818E **RESET MAXASSIGNED cannot be YES if the RESTART WITH value is changed.**

Explanation:
RESET MAXASSIGNED must be NO if the RESTART WITH value is changed.

System action:
Processing stops.

User response:
Set RESET MAXASSIGNED to NO if you change RESTART WITH.

ADB819E **Specification of "BY ALL" is not currently supported.**

Explanation:
The Revoke Impact function does not support the "BY ALL" option.

System action:
Processing stops.

User response:
Use Revoke Impact if "BY ALL" is specified after Revoke Impact produces correct results.

ADB820I **An object name contains mixed case characters. CAPS OFF is set.**

Explanation:
Because the collection ID for this object is in mixed case, the CAPS setting is turned off to avoid capitalization. Any changes made to the collection ID must be made in the desired case.

System action:
Processing continues.

User response

No action is required.

Related information

CAPS

ADB821I **Some data fields are changed to CAPS *setting*.**

Explanation

The CAPS setting is changed as requested.

setting
ON or OFF

System action:
Processing continues.

User response

No action is required.

Related information

CAPS

ADB822W **The CAPS command is not allowed on this panel.**

Explanation:

The CAPS command is not allowed in the current context. The fields on this panel do not support mixed-case values.

System action:
Processing continues.

User response

No action is required.

Related information
CAPS

ADB823W **Valid values of applcompat are *APPLCOMPAT_values*, where 5nn is a numeric value that cannot be greater than the current Db2 function level (*current_FL*)**

Explanation

The specified APPLCOMPAT value is not valid, because it is higher than the current Db2 function level.

APPLCOMPAT_values
Valid APPLCOMPAT values.

current_FL
The current function level at which the Db2 subsystem is running.

System action:
Processing continues.

User response:
Specify a valid value for APPLCOMPAT.

Related information
APPLCOMPAT bind option (Db2 12 for z/OS)

ADB824E **Unsupported object type *type* found during processing of the SB line command.**

Explanation

A Db2 object type that is not yet supported or an invalid type character was encountered by the SB (show base objects) line command on the **Package Dependencies (ADB21KD)** panel.

type
A one-character code that represents the Db2 object type. These object type codes are from the BTYPE column of the Db2 catalog table SYSPACKDEP.

System action:
Processing stops.

User response:
Contact IBM Software Support to report the message.

Related information
SYSPACKDEP catalog table (Db2 12 for z/OS)

ADB825E **INVALIDONLY(YES) is invalid with PLANMGMTSCOPE(ALL)**

Explanation:
These FREE PACKAGE command options are not compatible.

System action:
Processing stops.

Programmer response:
Specify a valid combination of options.

ADB900E **Error condition. An unrecognized object type *object_type* was passed when virtual changes were applied.**

Explanation:
The object type is unrecognized. It is unlikely that this error will cause a problem.

System action:
Processing continues.

User response:
Contact IBM support to report the message.

ADB901E **An error occurred in the *program_name*. Return code = *return_code*.**

Explanation:
An error occurred in the specified program. The program cannot continue.

System action:
Processing stops.

User response:
Contact IBM support to report the message.

ADB903I **The pending definition changes have been dropped.**

Explanation:
The pending DB2 definition changes have been dropped from the SYSPENDINGDDL table.

System action:
Processing continues.

User response:
No action is required.

ADB904E **The table *table_name* contains too many columns.**

Explanation:
You can assign up to 750 columns for a non-dependent table. Dependent tables can have up to 749 columns.

System action:
Processing stops.

User response:
Limit the number of columns to allowed values and try the operation again.

ADB906E **Export changes failed. Use TSO ISRDDN to view the ADBDIAG file contents and determine the cause of failure.**

Explanation:
The export changes procedure failed. Use TSO ISRDDN to check the ADBDIAG file contents. In the ADBDIAG file, you might find references to objects involved in the failed export changes procedure.

System action:
Processing stops.

User response:
Use TSO ISRDDN to check the ADBDIAG file contents. Review objects or messages in the file that indicate conflict.

ADB907E **The primary command is invalid. The valid primary command is *&validcmd.***

Explanation:
To add a product entry, use the primary command ADD. To update a product entry, use the primary command UPDATE or UPD. To delete a product entry, use primary command DELETE or DEL.

System action:
Processing stops.

User response:
Enter a valid value for the primary command.

ADB908E **Invalid buffer pool size. The buffer pool must be *&bpm.* and the size cannot be altered. To alter the buffer pool size to something other than &bpm, enter END to exit and return to the Table Spaces (ADB21S) panel. Then, use the line command ALT to redefine the table space. Do not use the AL command to change the buffer pool size to a different buffer pool size.**

Explanation:
The buffer pool size must be appropriate for the table space. If the buffer pool size of the table space is 4KB, the value of *&bpm.* is BP0-BP49, 8KB is BP8K0-BP8K9, 16KB is BP16K0-BP16K9, and 32KB is BP32K, BP32K1-BP32K9.

System action:
Processing stops.

User response:
Use the line command ALT to redefine the table space. Do not use the AL line command.

ADB909E **The Installation default parameters option is not available because Change Management was disabled at install time.**

Explanation:
The **Installation default parameters** option is not available because the Change Management database was not created or the CM option was disabled at install time. Db2 Admin Tool will use DB2 Utility default values instead.

System action:
Processing stops. The Db2 Admin Tool utility panels will allow you to specify the PARALLEL parameter according to the standard DB2 utility limits. See the DB2 Utility Guide and Reference for more information about the PARALLEL keyword.

User response

If the DB2 Utility default value limits are sufficient, then no action is needed.

If there is a need to enable the **Change installation default parameters** option on the Db2 Admin Tool Options panel (ADB2P), the Db2 Admin Tool administrator or installer should complete the following steps.

1. In Tools Customizer, navigate to the Customizer workplace: DB2 Admin Tool panel (CCQPWRK).
2. Issue the E line command for the Product parameters field.
3. On the Product parameters panel (CCQPPRD), scroll several pages to the Admin Tool setup task (create and upgrade) section, and enable the following options:
   - Change Management database - YES
   - Enable CM on Db2 Admin Tool primary menu - YES
4. Press PF3 to navigate back to the Customizer Workplace: DB2 Admin Tool panel (CCQPWRK).
5. Issue the G line command to regenerate the Admin Tool Setup Task job template ADBSETUP.
6. Submit the Admin Tool Setup Task job template ADBSETUP.
7. Submit the ADBBIND template.

ADB986E **An incorrect parameter, `<special register or session_variable>`, was specified for the default parameter, `<default>`. Use the**

U command to specify a valid `<parameter1>`.

Explanation:
An incorrect special register or session variable name has been specified for the default value.

System action:
Processing stops.

User response:
Either correct the default value, or use the U (update) command to change the special register or session variable with the GENERATED option.

ADB987E **The default value, `<default>`, is valid only on a column with `<data_type and length>` data type.**

Explanation:
You have specified an invalid data type or length for the data change operation, special register, or session variable used in the GENERATED expression.

System action:
Processing stops.

User response:
Correct the data type and /or length.

ADB988E **A special register or session variable must be specified. Use the U (update) command to specify one using the GENERATED option.**

Explanation:
A special register or session variable is missing for the GENERATED clause, as indicated by a default value of "a" or "b".

System action:
Processing stops.

User response:
Use the U (update) command to specify a special register or session variable with the GENERATED option.

ADB991E **The archive table cannot be defined as a parent or child in a referential constraint.**

Explanation:
You cannot specify an archive table that is defined as a parent or child in an existing referential constraint.

System action:
Processing stops.

User response:
Specify an archive table that is not defined as the parent or child in an existing referential constraint.

ADB992E **The archive-enabled table and the archive table must have the same *<parameter>*.**

Explanation:
The archive-enabled table and it's archive table must have the same encoding scheme and number of columns.

System action:
Processing stops.

User response:
Specify an archive table that has the same number of columns and the same encoding scheme as the archive-enabled table.

ADB993E **The *<parameter>* table must be the only table in the table space.**

Explanation:
In order to enable archiving, the specified table must be the only table in the table space.

System action:
Processing stops.

User response:
Specify a table that is the only table in the table space.

ADB994E **The *<parameter>* cannot include a SYSTEM_TIME or BUSINESS_TIME period.**

Explanation:
An archive-enabled or archive table cannot include a SYSTEM or BUSINESS time period.

System action:
Processing stops.

User response:
Specify a table that does not contain a period.

ADB995E **The *<parameter1>* table cannot include *<parameter2>*.**

Explanation

In order to enable archiving, neither the archive-enabled table nor the archive table can include any of the following:

- An identity, transaction-start-ID, row-begin, or row-end column
- A column mask or row permission

System action:
Processing stops.

User response:
Assure the archive and archive-enabled tables do not contain any of the above column attributes.

ADB996E **The *<parameter>* table cannot have an incomplete table definition.**

Explanation:
In order to enable archiving, the archive and archive-enabled tables must not have an incomplete table definition.

System action:
Processing stops.

User response:
Assure the tables are defined as complete.

ADB997E **The *<parameter>* table cannot contain a security label column.**

Explanation:
In order to enable archiving, neither the archive nor the archive-enabled table can contain a security label column.

System action:
Processing stops.

User response:
Assure the table does not contain a security label column.

ADB998E **The *<parameter>* table cannot be involved in a clone relationship.**

Explanation:
In order to enable archiving, neither the archive nor the archive-enabled table can be involved in a clone relationship.

System action:
Processing stops.

User response:
Assure the table is not involved in a clone relationship.

ADB999E **The archive table cannot be *<parameter>*.**

Explanation

You cannot specify as an archive table a view, a table implicitly created for an XML column, or any of the following:

- Clone table
- Global temporary table
- History table
- MQT
- Auxiliary table
- Existing archive table
- Archive-enabled table
- Catalog table

System action:
Processing stops.

User response:
Assure the table is not involved in a clone relationship.

ADB0014E **The input from the PARMS file is not valid. Comments are not allowed in the input file. The invalid input is *'text_that_is_invalid'*.**

Explanation:
The invalid input that is displayed in the message contains the text that most likely contains a comment.

System action:
Processing stops. Additional errors in the input are not reported.

User response:
Check the input file and verify that no comments exist.

ADB0015E **The input from the PARMS file is not valid. A parameter name might be misspelled. The invalid input is *'text_that_is_invalid'*.**

Explanation:
The invalid input that is displayed in the message contains the text that likely contains a misspelled parameter name.

System action:
Processing stops. Additional errors in the input are not reported.

User response:
Check the input file and verify that all the parameter names are spelled correctly.

ADB0016E **The input from the PARMS file is not valid. The first character of the invalid input is *first_character* and the hexadecimal value of this character is *hexadecimal_value_of_first_character*. If the character is not displayed, check the hexadecimal value. The invalid input is *'text_that_is_invalid'*.**

Explanation:
A character was detected in a location in the file that is not allowed by the parameter syntax.

System action:
Processing stops. Additional errors in the input are not reported.

User response:
Verify input and try again.

Related concepts

"Parameters for CM batch interface" on page 662

The Change Management (CM) batch interface includes a list of parameters that enable you to control various aspects of managing changes, including what action the CM batch interface performs when called.

ADB0017E **An error occurred while reading the input parameters from the PARMS file. The invalid input is *'text_that_is_invalid'*.**

Explanation

The exact cause of this error is unknown. The most likely cause is unmatched escape characters for a parameter value. A parameter value must be enclosed with the escape character, which is an apostrophe (').

Remember: Two consecutive escape characters must be used to represent one escape character within a parameter value.

Here is an example of an invalid and a valid use of escape characters:

- Invalid: job_card_line_1 = '//TEST1234 JOB (INFO),'TEST''
- Valid: job_card_line_1 = '//TEST1234 JOB (INFO),''TEST'''

System action:
Processing stops. Additional errors in the input are not reported.

User response:
Verify input and try again.

ADB0380E **Module *module_name* - Severe error. *program_name* is halted.**

Explanation:
The specified module has encountered a severe problem and the specified program has halted.

System action:
A return code of 12 is set and processing stops.

User response:
An internal error has been detected. Contact IBM Software Support.

ADB1003E **An error occurred while processing DBname= *requested_database*, TSname= *requested_table_space*.**

Explanation:
An unexpected and unknown processing error occurred. The most recent database or table space that was requested is displayed.

System action:
Processing stops.

User response:
Look for other messages in the job output that might indicate the cause of the error. Contact IBM Software Support if needed.

ADB1011E **Error in input parameter *parameter-name*. Processing cannot continue. Possible bad option data, misspelled option name, or other form of invalid syntax.**

Explanation

The listed ADB2RE parameter is not valid.

parameter-name
The name of the input parameter that is not valid.

System action:
Processing stops.

User response

Check the input parameter as follows:

- Check the spelling of option names.
- Check that option values are correct and enclosed in single quotes.
- Ensure that options are separated by commas.
- Ensure that the entire parameter string is terminated by a semicolon.

You can also use the DEBUG option to help you troubleshoot the parameter. When you specify DEBUG, the entire parameter string is written to the SYSPRINT file in the WLM job.

After correcting the parameter, try again.

ADB1012W **Error in input parameter *parameter-name*. Defaults will be used. Possible bad option data, misspelled option name, or other form of invalid syntax.**

Explanation

Because the listed ADB2RE parameter is not valid, it will be ignored. The default value of this parameter will be used instead.

parameter-name
The name of the input parameter that is not valid.

System action:
Processing continues.

User response

If you do not want the default value to be used, correct the input parameter and try again. You can check the input parameter as follows:

- Check the spelling of option names.

- Check that option values are correct and enclosed in single quotes.
- Ensure that options are separated by commas.
- Ensure that the entire parameter string is terminated by a semicolon.

You can also use the DEBUG option to help you troubleshoot the parameter. When you specify DEBUG, the entire parameter string is written to the SYSPRINT file in the WLM job.

ADB1026E **The parameter input file is empty.**

Explanation:
The parameter input file is generated by the product.

System action:
Processing stops.

User response:
If the JCL job step that contains the empty parameter file was generated by the product, contact IBM Software Support.

ADB1031E **DDL cannot be generated for DB2 release *requested_DB2_release*. Supported releases are *mininum_supported_DB2_release* through *maximum_supported_DB2_release* .**

Explanation:

System action:
Processing stops.

User response:
Specify a supported DB2 release and try again.

ADB1032E **DDL cannot be generated for DB2 release *local_DB2_system_release*. Supported releases are *minimum_supported_DB2_release* through *maximum_supported_DB2_release* .**

Explanation:

System action:
Processing stops.

User response:
Ensure that a DB2 connection exists to a supported DB2 release.

ADB1187E **The exclude specification *exclude_specification_owner* *exclude_specification_name* does not exist.**

Explanation:
A user-specified exclude specification was not found.

System action:
Processing stops.

User response:
Ensure that the specified owner and name are correct.

ADB1223E ***module_name* Unexpected sqlcode in: *error_function***

Explanation:
The specified module received an unexpected SQL return code from DB2.

System action:
Processing stops.

User response:
See the details for the SQL code in the DB2 documentation.

ADB1241E **An unexpected error occurred while processing version scope *version_scope_qualifier*.*version_scope_name*. Reason code=*reason_code***

Explanation:
Reason codes: 1,3 - Report this error to IBM. 2,4 - Look for other error messages to determine the cause.

System action:
Processing stops.

User response:
Check the reason code and take the indicated action.

ADB1426E **An internal error occurred. Table *table_creator*.*table_name* could not be found in an internal data storage.**

Explanation:

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1429W **Clone table *clone_schema* *clone_name* required that base table *base_table_schema* *base_table_name* exist before the clone can be created.**

Explanation:
The GEN function created DDL to add a clone, but the base table is not part of the DDL.

System action:
None.

User response:
No action is necessary if you do not want the base table included in the DDL. Otherwise, include the base

table *base_ table_schema base_ table_ name* and run GEN again.

ADB1456e **The number of plan dependencies has exceeded the product limit of 32K.**

Explanation:

System action:
No system action is taken.

User response:
A product limit has been reached. The maximum number of plan dependencies for each plan is 32K. Processing stops.

ADB1457e **The number of package dependencies has exceeded the product limit of 32K.**

Explanation:

System action:
No system action is taken.

User response:
A product limit has been reached. The maximum number of package dependencies for each package is 32K. Processing stops.

ADB1458e **The number of packages has exceeded the product limit of 32K.**

Explanation:

System action:
No system action is taken.

User response:
A product limit has been reached. The maximum number of packages that can be generated is 32K. Processing stops.

ADB1602E **No SYSVOLUMES record was found in the DB2 catalog for STOGROUP *stogroup_name*.**

Explanation:

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1603W **Stogroup not found for DBname= *database_name*, SGname= *stogroup_name***

Explanation

The listed storage group was not found in the Db2 catalog.

database_name
The database name.

stogroup_name
The storage group name.

System action:
No system action is taken.

User response:
Specify an existing storage group.

ADB1604W **Database not found, DBname= *database_name***

Explanation

The listed database was not found in the Db2 catalog.

database_name
The database name.

System action:
No system action is taken.

User response:
Specify an existing database.

ADB1605I **No Table spaces for Database= *database_name***

Explanation

No table spaces were found for the listed database in the Db2 catalog.

database_name
The database name.

System action:
No system action is taken.

User response:
Specify a database with at least one table space.

ADB1606W **Table space not found, DBname= *database_name* TSname=*table_space_name***

Explanation

The listed table space was not found in the Db2 catalog.

database_name
The database name.

table space_name
The table space name.

System action:
No system action is taken.

User response:
Specify an existing table space.

ADB1607E **A SYSDATABASE record was not found for table space *table_space_name*, database*database_name*.**

Explanation:
The database name recorded in the SYSTABLESPACE record for the specified table space does not have a SYSDATABASE record in the DB2 catalog.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1609W ***module-name object-type object-name*** **not found.**

Explanation

The specified object was not found.

module-name
The module name is listed when this message is included in the ADBMSGS data set.

object-type
The object type, such as table, view, alias, or synonym.

object-name
The qualified name of the object.

System action:
Processing continues.

User response:
Verify whether this message is expected. If you expected these objects to be found, make changes as needed.

Related tasks

"Consolidating messages into a single file" on page 246
Some batch interface jobs produce multiple output files, each of which contain a number of messages. To simplify troubleshooting, you can consolidate these messages into one file, or data set. You can do this consolidation for batch jobs for functions such as Change Management (CM) batch interface, ALT, Compare, and ADBTEP2.

ADB1610E **A table space was not found:** ***database_name.table_space_name***

Explanation:
The SYSTABLESPACE record for the specified table space was not found in the DB2 catalog.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1613E **The table associated with an index was not found. The index name is** ***index_name*****. The table name is** ***table_name*****.**

Explanation:
The SYSTABLES record for the table name recorded in a SYSINDEXES record was not found in the DB2 catalog.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1614E **The database associated with an index was not found. The index name is** ***index_name*****. The database name is** ***database_name*****.**

Explanation:
The SYSDATABASE record for the database name recorded in a SYSINDEXES record was not found in the DB2 catalog.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1627E ***ADB*****2GEN - Location** ***location_name*** **is not defined on the local DB2 system.**

Explanation:

System action:
Processing stops.

User response:
Ensure that the DB2 location name is correct.

ADB1628E ***program_name*** **- Location** ***location_name*** **is not a DB2 for z/OS system. Generate DDL will not work for this location.**

Explanation:
The specified program is only supported to run on a DB2 for z/OS system.

System action:
Processing stops.

User response:
Ensure that the specified location is a DB2 for z/OS system.

ADB1636E **An internal limit has been reached. The catalog row stack is full.**

Explanation:

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1639E An internal error has occurred. An unexpected DB2 catalog row type of *row_type* was requested.

Explanation:

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1646E An error has occurred while generating DDL for an object.

Explanation:

System action:
Processing stops.

User response:
Look for error messages prior to this message for additional information.

ADB1650E An error has occurred while generating the storage group for database *database_name*.

Explanation:

System action:
Processing stops.

User response:
Look for error messages prior to this message for additional information.

ADB1651E An error has occurred while generating the storage group for table space *database_name.table_space_name* .

Explanation:

System action:
Processing stops.

User response:
Look for error messages prior to this message for additional information.

ADB1652E An error has occurred while generating the storage group for index *index_schema.index_name*.

Explanation:

System action:
Processing stops.

User response:
Look for error messages prior to this message for additional information.

ADB1653E Storage group *stogroup_name* was not found in the DB2 catalog.

Explanation:
A storage group that is associated with a table space or index was not found in the DB2 catalog.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1658W Index *index_creator_v index_name_v* is being generated because the ROWID column *ROWID_column_name_v* on table *table_creator_v table_creator_name_v* will be converted from GENERATED ALWAYS to GENERATED BY DEFAULT. Converting the ROWID to GENERATED BY DEFAULT is done to allow the ROWID table data to be loaded back into the table using the DB2 LOAD utility.

System action:
None. GEN processing continues.

User response:
None.

ADB1660W The database was skipped because a temporary database is not supported in DB2 V9 or later versions.

Explanation:
A temporary database is being generated for DB2 9 function mode, but the DB2 9 function mode does not support temporary databases. The GEN function will not generate DDL for the temporary database.

System action:
None. GEN processing continues.

User response:
No action is required.

ADB1661W Table space *database table_space* was skipped because it was implicitly created.

Explanation:
The GEN functon does not generate information for an implicit table space for XML columns.

System action:
None. GEN processing continues.

User response:
No action is required.

ADB1662W Table *table_creator table_name* was skipped because it is an implicit table that was created for XML columns.

Explanation:
GEN does not generate information for an implicit table space that was created for XML columns.

System action:
None. GEN processing continues.

User response:
No action is required.

ADB1663W The owner of *object_type qualified_object_name* is a role.

Explanation:
If the object owner should be a role when the object is created, a trusted context must be established when creating the object.

System action:
None. GEN processing continues.

User response:
Establish a trusted context to create the object with a role as the object owner. You can ignore this message if you do not want a role as the object owner.

ADB1664E An internal error occurred. Diagnostic text= *diagnostic information for IBM optional object type optional object qualifier. optional object name optional additional diagnostic text optional additional diagnostic text*.

Explanation:
This message is issued for several types of internal errors.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1666W A SYSAUXRELS row was not found for the DB2 auxiliary table *aux_tbcreator*.*aux_tbname*. The DB2 table space will not be generated.

Explanation:
If a row is not found in the SYSAUXRELS catalog table, the relationship between the base table and the auxiliary table is unknown and GEN will not generate the table space of the auxiliary table.

System action:
GEN processing continues.

User response:
No action is required.

ADB1668W Multiple accelerator aliases can be used to create the high availability accelerator-only table (HA AOT) *accelerator-only-table-name*. HA AOT *accelerator-only-table-name* will be created with the accelerator alias *accelerator-alias-name*.

Explanation

Multiple accelerator group names, or *aliases*, were found for the specified accelerator-only table (AOT). However, the generated DDL will include only the first accelerator alias that was returned. This name is listed at the end of the message.

accelerator-only-table-name
The qualified name of the AOT.

accelerator-alias-name
The accelerator alias name that will be used in the generated DDL. The accelerator alias name represents multiple accelerators that are defined with HA AOT.

System action:
Processing continues.

User response

No action is required.

ADB1669W No accelerator alias includes *list-of-accelerator-names* for the high availability accelerator-only table (HA AOT) *accelerator-only-table-name*. DDL is not generated for HA AOT *accelerator-only-table-name*.

Explanation

Db2 Admin Tool could not find an accelerator group, or *alias*, that includes the specified accelerators. Therefore, DDL is not generated for the specified accelerator-only table (AOT).

list-of-accelerator-names
The names of the accelerators that must be included in the accelerator group.

accelerator-only-table-name
The qualified name of the AOT.

System action:
Processing continues.

User response:
Create an accelerator alias that includes only the listed accelerators (in *list-of-accelerator-names*).

ADB1670W **DDL is not generated for *object-type object-name* because it is not supported on the function level *target-function-level*. The required function level is *required-function-level*.**

Explanation

The requested DDL cannot be generated, because the target function level does not support the required syntax.

object-type
The type of Db2object.

object-name
The qualified name of the object.

target-function-level
The function level to be used when generating the DDL.

required-function-level
The minimum function level required to support the DDL.

System action:
Processing continues.

User response

Run the GEN command again. On the resulting **Generate SQL from DB2 catalog (ADB2GENB)** panel, set the following fields so that the target version and function level is *required-function-level* or higher:

- **Target DB2 version**
- **Target Function Level**

The current Db2 version and target function level are listed next to these fields. These values are used if the fields are left blank.

Related information

"Target function level" on page 72

ADB1816E **A procedure parameter data type of *data_type_id* is not yet supported.**

Explanation:
An unsupported data type was found for a procedure parameter.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1837E **The value for DSSIZE of a table space is not yet supported.**

Explanation:

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1841E **A function parameter data type of *data_type_id* is not yet supported.**

Explanation:
An unsupported data type was found for a function parameter.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1847E **A parser error has occurred for the following *statement_type*. GEN cannot complete the request.**

Explanation:
The statement could not be parsed by the DB2 Admin parser. Because the GEN request contained DDL changes (such as masking, change owner, change schema, RUN sqlid, and so on), processing stops. The unformatted DDL is generated as an SQL comment.

System action:
Processing stops.

User response:
Run GEN again with no DDL changes. If the parser error still occurs then contact IBM Software Support. If the parser error does not occur then ensure that the DDL changes are correct.

ADB1871E **An internal limit has been reached. The DDL stack is full.**

Explanation:

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1873E **Processing ended but not all supplied catrows were used.**

Explanation:

System action:
Processing continues.

User response:
Contact IBM Software Support.

ADB1875E **An unexpected error return code was received while a mask was being processed.**

Explanation:

This error can be an internal error or can be caused by an invalid mask being specified.

System action:
Processing stops.

User response:
If this message was caused by an invalid mask being specified, it will be preceded by additional related messages. Refer to those messages to attempt to correct the problem. If this message is an internal error (that is, is not preceded by additional related messages), contact IBM Software Support.

ADB1877E **An error occurred in the DB2 Admin auth-switching module, RC=*return_code***

Explanation:

System action:
Processing stops.

User response:
If this message is preceded by additional related messages, refer to those messages for more details about this error condition. If this message is not preceded by additional related messages, contact IBM

ADB1907E **An invalid TYPE value of *invalid_type_value* was specified for the*program_name* program.**

Explanation:

System action:
Processing stops.

User response:
If the TYPE parameter was built by the product, contact IBM Software Support. Otherwise, ensure the value for TYPE matches a supported value as documented in the DB2 Admin Users Guide.

ADB1915W **The original DDL for the following object will be generated as it is stored in DB2. Verify the DDL.**

Explanation:
The internal DDL buffer of the DDL statement the GEN program attempted to create exceeded 2 MB. Since the GEN request did not contain any DDL changes (such as masking, change owner, change schema, or Run SQLID) the original DDL that is stored in DB2 is generated.

System action:
None.

User response:
Verify the DDL is correct.

ADB1916E **The DDL for the following object cannot be created within the 2 MB limit. GEN cannot complete the request.**

Explanation:

System action:
No system action is taken.

User response:
GEN processing stops. The DDL statement the GEN program attempted to create exceeded the output buffer size. The GEN program will not attempt to generate the original DDL stored in DB2. This is most likely because of at least one of the following: - a request was made to change the DDL (i.e. masking, change owner, change schema, RUN sqlid, etc.) - the object was originally created using an ALTER statement - the object has a table parameter GEN cannot complete the request. Try running GEN again with no DDL change requests.

ADB1917W **Unformatted DDL will be generated for the following object because of an unknown formatter error. Verify the DDL.**

Explanation:
An unknown internal formatter error occurred. Since the GEN request did not contain any DDL changes (such as masking, change owner, change schema, or Run SQLID), the unformatted DDL is generated..

System action:
None.

User response:
Verify the DDL is correct.

ADB1918E **An unknown formatter error occurred. GEN cannot complete the request for the following *stmt_type*.**

Explanation:

System action:
No system action is taken.

User response:
An unknown internal formatter error occurred. Since the GEN request contained DDL changes (i.e. masking, change owner, change schema, RUN sqlid, etc.), processing stops. GEN cannot complete the request. Try running GEN again with no DDL changes. The unformatted DDL is generated but as an SQL comment.

ADB1919W **Unformatted DDL will be generated for the following object because the formatted DDL exceeded 2 MB. Verify the DDL.**

Explanation:
The output formatter buffer size was exceeded. Since the GEN request did not contain any DDL changes

(such as masking, change owner, change schema, or Run SQLID), the unformatted DDL is generated.

System action:
None.

User response:
Verify the DDL is correct.

ADB1920E The formatted DDL has exceeded 2 MB. GEN cannot complete the request for the following *stmt_type*.

Explanation:

System action:
No system action is taken.

User response:
The output formatter buffer size was exceeded. Since the GEN request contained DDL changes (i.e. masking, change owner, change schema, RUN sqlid, etc.), processing stops. GEN cannot complete the request. Try running GEN again with no DDL changes. The unformatted DDL is generated but as an SQL comment.

ADB1921W The current APPLCOMPAT or target function level is lower than 507. Generate CREATE PROCEDURE statement instead.

Explanation

The requested CREATE OR REPLACE PROCEDURE statement could not be generated, because the current APPLCOMPAT value or target function level is lower than 507. A CREATE PROCEDURE statement was generated instead.

System action:
Processing continues.

User response:
No action is required. However, if you want a CREATE OR REPLACE PROCEDURE statement, change the APPLCOMPAT value or target function level to 507 or later and regenerate the DDL.

Related concepts

"Db2 function level settings in Db2 Admin Tool" on page 70
Within Db2 Admin Tool, you can specify the Db2 function level, the APPLCOMPAT function level, the target function level, and the maximum Db2 function level accepted.

ADB1922W The native stored procedure version must be V1 to generate the CREATE OR REPLACE PROCEDURE with SPECIFIC clause.

Explanation

The CREATE OR REPLACE PROCEDURE statement with SPECIFIC clause could not be generated. The version identifier for the native stored procedure must be V1 to include the SPECIFIC clause. A CREATE PROCEDURE statement was generated instead.

System action:
Processing continues.

User response

No action is required.

ADB1923W View *view-name* APPLCOMPAT *applcompat-value* IS HIGHER THAN TARGET FL *target-applcompat-value*. SET VIEW APPLCOMPAT TO *target-applcompat-value*

Explanation

The generated DDL for the specified view includes a SET CURRENT APPLICATION COMPATIBILITY statement. Because the current APPLCOMPAT function level is higher than the target function level, the target function level was used in the SET CURRENT APPLICATION COMPATIBILITY statement.

view-name
The first twelve characters of the view name.

applcompat-value
The value of the CURRENT APPLICATION COMPATIBILITY special register.

target-applcompat-value
The target application compatibility value.

System action:
Processing continues.

User response

No action is required.

Related information

"APPLCOMPAT function level" on page 71

"Target function level" on page 72

ADB1933E The DB2 Admin parser could not parse a statement. An SQL comment containing the original DDL will be generated.

Explanation:
The DDL statement that the GEN program attempted to create encountered a parser error. GEN cannot complete the request.

System action:
Processing stops.

User response:
Look for other messages that identify the object being parsed. Try running GEN again with no DDL change requests.

ADB1935E **SQL body not found. GEN cannot complete the request. An SQL comment containing the original DDL will be generated.**

Explanation:
The DB2 Admin parser could not locate the SQL body in the original DDL text. The GEN program will not attempt to generate the original DDL stored in DB2. This is most likely because one or more of the following: - A request was made to change the DDL, for example, masking, change owner, change schema, and RUN sqlid. - The object was originally created using an ALTER statement. - The object has a table parameter.

System action:
Processing stops.

User response:
Try running GEN again with no DDL change requests. Contact IBM Software Support if needed.

ADB1943E **The "Only" value cannot be specified for both the "Generate catalog stats" and "Include DB2 pending chgs" options.**

Explanation:
Choosing "Only" for the specified options is mutually exclusive.

System action:
Processing stops.

User response:
Specify "Only" for one of the identified options but not both.

ADB1944E **The SYSTABLEPART table contains a record of PARTITION *part_num* of *obj_type obj_qual.obj_name*, which has an invalid value "*err_value*" for part *err_seqno* of column LIMITKEY.**

Explanation:
An attempt was made to process the value of a limit key but an unexpected and presumed invalid value was encountered.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB1945W **The INLINE LENGTH *length* clause for the column *column_name* in table *table_name* is not generated because the zparm SPRMRRF is set to disable.**

Explanation:
The DB2 zparm SPRMRRF is set to disable. When zparm SPRMRRF is disabled, INLINE LENGTH clauses for columns are not generated.

System action:
Processing continues.

User response:
Add INLINE LENGTH *length* clauses manually, if needed.

ADB1950E **The "Only" value cannot be specified for both the "Generate index cleanup" and "Include DB2 pending chgs" options.**

Explanation:
Choosing "Only" for the specified options is mutually exclusive.

System action:
Processing stops.

User response:
Specify "Only" for one of the identified options but not both.

ADB1951E **An error occurred when the Gen component called the ADB2ZP program to get the DB2 system parameter (DSNZPARM) values.**

Explanation:

System action:
No system action is taken.

User response:
See the error that was written in the log file by the ADB2ZP program. Resolve the problem and retry.

ADB1952W **An error occurred when the Gen component called the ADB2ZP program to get the DB2 system parameter (DSNZPARM) values.**

Explanation:
However, the DSNZPARM values are not needed because no request was made to remove the default values or generate ADMIN ALTER IMPLICIT statements.

System action:
The error is ignored and processing continues.

User response:
See the error that was written in the log file by the ADB2ZP program. Resolve the problem and retry.

ADB1953E **ALTER TABLE DROP COLUMN statements were generated for DB2 pending changes. These statments cannot be run on the specified DB2 level. All DDL statements are generated but GEN ends with RC=12.**

Explanation:
A DROP COLUMN DB2 pending change exists and a value other than "No" was specified for the "Include DB2 pending changes" option. This results in an ALTER TABLE DROP COLUMN statement being generated that is not supported on the DB2 level specified for the "Target DB2 version" option.

System action:
All DDL is generated but GEN ends with RC=12.

User response:
To avoid this condition, specify "Target DB2 version" 1115 or higher, or complete or DROP the DB2 pending changes before running GEN.

ADB1956E **An unsupported ARRAYINDEXTYPE value (*arrayindextypeid_value*) was found in a SYSDATATYPES record.**

Explanation:
The value ARRAYINDEXTYPE is not supported.

System action:
Processing stops.

User response:
Verify that the version of GEN is supported on this version of DB2 and that the value of ARRAYINDEXTYPE is valid.

ADB1957E **The option GETDB2ZP='N' is specified; therefore GEN cannot call the DB2 stored procedure ADMIN_INFO_SYSPARM and get the DB2 system parameter (DSNZPARM) values. The DSNZPARM values are required when GEN generates a version file.**

Explanation:
The GEN function of Db2 Admin Tool cannot get the Db2 subsystem parameter values (in the Db2 DSNZPARM initialization parameter module). These DSNZPARM values are needed by GEN to write a version file. These DSNZPARM values are also required by subsequent functions.

System action:
Processing stops.

User response:
On the **Admin Defaults (ADB2P2)** panel, specify YES in the **Get DB2 ZPARM** field.

ADB1958W **The option GETDB2ZP='N' is specified; therefore GEN cannot call the DB2 stored procedure ADMIN_INFO_SYSPARM and get the DB2 system parameter (DSNZPARM) values. The following DSNZPARM values will be used when removing DDL default values and generating ADMIN ALTER IMPLICIT statements: TBSBPOOL=BPP0; TBSBP8K=BP8K0; TBSBP16K=BP16K0; TBSBP32K=BP32K; TBSBPLOB=BP0;TBSBPXML=BP16K0; IDXBPOOL=BP0; WLMENV=''; PADIX=NO; IMPTSCMP=NO; LOB_INLINE_LENGTH=0; IMPTSCMP = NO; MAX_UTIL_PARTS=''; RRF=TRUE.**

Explanation:
The GEN function of Db2 Admin Tool is to use the listed default values for Db2 subsystem parameters, because it cannot get those values from Db2 (in the Db2 DSNZPARM initialization parameter module). These DSNZPARM values are needed by GEN to remove DDL default values and to generate ADMIN ALTER IMPLICIT statements.

System action:
Processing continues.

User response:
You need to take action only if you want GEN to use the Db2 subsystem parameter values instead of the default values. In this case, on the **Admin Defaults (ADB2P2)** panel, specify YES in the **Get DB2 ZPARM** field.

ADB1959W **DB2 stored procedure ADMIN_INFO_SYSPARM call to get the DB2 system parameters (DSNZPARM) failed. The default value of APPLCOMPAT DB2 system parameter is used as V11R1 for DB2 V11 new function mode and V12R1 for DB2 V12 new function mode.**

Explanation:
Db2 Admin Tool was unable to get the Db2 subsystem parameter values (in the Db2 DSNZPARM initialization parameter module). Therefore, Db2 Admin Tool is to use the default value for the APPLCOMPAT subsystem parameter.

System action:

Processing continues.

User response

No action is required.

Related reference

APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter) (Db2 12 for z/OS)

APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter) (Db2 11 for z/OS documentation)

ADB1960W **The option GETDB2ZP='N' is specified; therefore GEN cannot call the DB2 stored procedure ADMIN_INFO_SYSPARM and get the DB2 system parameter (DSNZPARM) values. The default value for APPLCOMPAT DB2 system parameter is used as V11R1 for DB2 V11 new function mode and V12R1 for DB2 V12 new function mode.**

Explanation:
The GEN function of Db2 Admin Tool is to use the default value for the APPLCOMPAT subsystem parameter, because it cannot get the value from Db2 (in the Db2 DSNZPARM initialization parameter module).

System action:
Processing continues.

User response:
You need to take action only if you want GEN to use the subsystem's value for APPLCOMPAT instead of the default value. In this case, on the **Admin Defaults (ADB2P2)** panel, specify YES in the **Get DB2 ZPARM** field.

Related reference

APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter) (Db2 12 for z/OS)

APPL COMPAT LEVEL field (APPLCOMPAT subsystem parameter) (Db2 11 for z/OS documentation)

ADB1961I **The source for EXTERNAL SQL procedure *procschema.procname* was not found in SYSROUTINES_SRC. The load module may need to be copied manually to the target library and a BIND of the DBRM may be required.**

Explanation

The external SQL procedure was not found in the Db2 catalog table SYSIBM.SYSROUTINES_SRC. Therefore, the routine body is not included in the generated DDL.

procschema
The schema for procedure.

procname
The name of the external SQL procedure.

System action:
Processing continues.

User response:
If necessary, copy the load module for the program to the target library and bind the DBRM.

ADB1970W **Db2 function level *function-level* is tolerated but not supported by this function. New attributes or objects for this Db2 function level might not be handled or displayed correctly. The maximum supported function level is *supported-function-level*.**

Explanation

This function of Db2 Admin Tool can run with the current Db2 version function level. However, this function does not support all new enhancements in the function level. If you use any enhancements in the function level, affected objects or attributes might not be handled or displayed correctly. If you do not use new enhancements in the function level, this function runs normally.

function_level
The function level of the Db2 subsystem or member on which you are running Db2 Admin Tool.

supported-function-level
The highest function level that is supported by this function of Db2 Admin Tool.

System action:
Processing continues.

User response:
Consider the implications of running Db2 Admin Tool with the listed Db2 function level and determine whether to proceed.

Related concepts

“Support for Db2 continuous delivery” on page 69

Db2 12 for z/OS introduced the concept of a function level to support continuous delivery of new enhancements. A *function level* is a single PTF that enables the activation of a specific set of Db2 enhancements. Db2 Admin Tool support for each function level is delivered in the maintenance stream.

“Db2 function level settings in Db2 Admin Tool” on page 70
Within Db2 Admin Tool, you can specify the Db2 function level, the APPLCOMPAT function level, the target function level, and the maximum Db2 function level accepted.

ADB1971S **To continue using this unsupported function level, you must accept the risk. Specify the "Max Db2 function level accepted" on the Admin Defaults panel or batch parameter ACCEPT_FL=’*xxx*’ in the options for this function.**

Explanation

The current Db2 version function level is not tolerated by this function.

xxx
The 3-digit number for the Db2 version function level.

System action:
Processing stops.

User response

If you want to continue with the current function level, you must specify it as the *maximum Db2 function level accepted* as follows:

- For CM batch or GEN jobs, specify the accept_fl parameter
- For any other functionality, specify the **Max Db2 function level accepted** field on the **Admin Defaults (ADB2P2)** paneland rerun the function.

Related concepts

“Support for Db2 continuous delivery” on page 69
Db2 12 for z/OS introduced the concept of a function level to support continuous delivery of new enhancements. A *function level* is a single PTF that enables the activation of a specific set of Db2 enhancements. Db2 Admin Tool support for each function level is delivered in the maintenance stream.

“Db2 function level settings in Db2 Admin Tool” on page 70
Within Db2 Admin Tool, you can specify the Db2 function level, the APPLCOMPAT function level, the target function level, and the maximum Db2 function level accepted.

Related reference

“CM batch parameter definitions” on page 664
You can use Change Management (CM) batch interface parameters to control Change Management (CM) actions and settings.

ADB2RE stored procedure (for GEN parameters)
The ADB2RE stored procedure generates SQL for objects from the Db2 catalog.

ADB1972W **Db2 function level *function-level* is not tolerated by this function. Unpredictable errors may occur and new attributes or objects for this Db2 function level might not be handled or displayed correctly. The maximum tolerated function level is *tolerated-function-level*.**

Explanation

This function of Db2 Admin Tool does not tolerate the current Db2 version function level. If you continue, unpredictable errors might occur.

function_level
The function level of the Db2 subsystem or member on which you are running Db2 Admin Tool.

tolerated-function-level
The highest function level that is tolerated by this function of Db2 Admin Tool.

System action:
Message ADB1971S is displayed and processing stops.

User response:
Decide whether you want to continue with the listed Db2 function level.

Related concepts

“Support for Db2 continuous delivery” on page 69
Db2 12 for z/OS introduced the concept of a function level to support continuous delivery of new enhancements. A *function level* is a single PTF that enables the activation of a specific set of Db2 enhancements. Db2 Admin Tool support for each function level is delivered in the maintenance stream.

ADB1973W **Index *index_name* is a spatial index, which is not supported.**

Explanation:
The index that you are attempting to compare is a spatial index. Object Compare does not support spatial indexes. Therefore, this index will not be processed or displayed correctly.

System action:
Processing continues.

User response:

None.

ADB1974E **Package *package-name* has qualifier SYSIBM. Using this package from a catalog copy would have resulted in DDL based on the local Db2 system catalog. GEN processing stops.**

Explanation

Db2 Admin Tool is currently using a copy of the catalog. However, because the indicated package has the qualifier SYSIBM, any generated DDL would be based on the Db2 catalog, not the copy.

package-name
The package name.

System action:
Processing stops.

User response:
Rebind the packages for the catalog copy with the same qualifier as the catalog copy owner. To do this rebind, execute the BIND*x* step of the DDLBND*xx* job by manually editing the RESTART parameter in the job card.

Related tasks

"Making copies of the Db2 catalog for Db2 Admin Tool" on page 1049

ADB2*nnn*I **Parameter name: *name*. Valid values: *values***

Explanation:
Each ADB2*nnn*I message (where *nnn* is a 3-digit number) lists valid values for the specified parameter.

User response

No action is required.

ADB3000E **An error occurred while processing the *object_name* object in the statement type of *stmt_type*. Object already exists.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are using DB2 Object Comparison Tool to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3000I **Default message - At least one message must exist in a message module. This message can be changed.**

Explanation:
This is a comment that can be used to explain the message. This comment tag is optional and is not displayed with the message.

System action:
This is a comment that can be used to explain the system action. This tag is optional and is not displayed with the message.

User response:
This is a comment that can be used to explain the programmer response. This tag is optional and is not displayed with the message.

ADB3001E **An error occurred while processing the *object_name* object in the statement type of *stmt_type*. Object does not exist.**

Explanation

When validating the syntax of the generated SQL statements for consistency, Db2 Admin Tool found an error with a statement of the type listed. The statement references an object that does not exist.

If the object existed in the catalog but was dropped by one of the SQL statements in the DDL, the message also includes the following text:

```
Object was dropped and not recreated.
```

If this drop was implicit, this message text includes the word `implicitly`, as follows:

```
Object was implicitly dropped and not recreated.
```

object_name
The type and name of the object in the statement.

stmt_type
The type of SQL statement, such as ALTER.

System action:
After the validate report is generated, processing stops with return code 8.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with Db2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3002E **An error occurred while processing the *object_name* object in the statement type of *stmt_type*. The**

object *object_name_2* does not exist.

Explanation

When validating the syntax of the generated SQL statements for consistency, Db2 Admin Tool found an error with a statement of the type listed. The statement creates or processes an object that references a second object that does not exist.

If the object was dropped by one of the SQL statements in the DDL, the message also includes the following text:

```
Object was dropped and not recreated.
```

If this drop was implicit, this message text includes the word `implicitly`, as follows:

```
Object was implicitly dropped and not recreated.
```

object_name
The type and name of the object in the statement.

stmt_type
The type of SQL statement, such as CREATE.

object_name_2
The type and name of the object referenced by *object_name*.

System action:
After the validate report is generated, processing stops with return code 8.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with Db2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3003E **An error occurred while processing the *object_name* object in the statement type of *stmt_type*. A clustering index already exists on *object_name2*.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3004E **An error occurred while processing object name *object_name* in statement type *statement_type*. The object was dropped many times.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. The object was dropped many times.

System action:
Processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are using DB2 Object Comparison Tool to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3004W **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. Multiple attempts were made to drop the object, but the object cannot be dropped.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing continues.

User response:
If you are validating a WSL, correct the SQL statement in the work statement list (WSL) and re-validate the WSL. If you are comparing objects with DB2 Object Comparison, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3005E **An error occurred while processing the *object_name* object in the statement type *stmt_typ*. The object and a foreign key must exist.**

Explanation:
The foreign key for the object cannot be found. The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:

Ensure that the object and foreign key for the object exists. If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3006E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The column *obj_name2* does not exist in the table.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3007E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The column *obj_name2* is not part of the parent table primary key.

Explanation:
The column that is referenced against the parent table primary key does not exist. The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3008E An error occurred while processing the *obj_name* object in the statement type of *stmt_type*. The referenced key has been dropped.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate idate Report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects withDB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3009E An error occurred while processing the *obj_name* object in the statement type of *stmt_type*. The number of index partitions does not match the number of table space partitions.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3010E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The referenced column *obj_name* does not exist in the parent table.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3011E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The table space is partitioned but a partitioning index has not been found.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement..

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool , correct the SQL statement in the DDL source and re-generate the compare job.

ADB3012E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The primary index or the index that is enforcing unique constraint does not have a matching primary or unique key.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3013E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The primary key or unique key does not have a matching primary index or index enforcing unique constraint.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3014E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The column *obj_name2* does not exist in the table or the table does not exist, nor is the column name a known global variable..

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3015E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The object *obj_name2* does not exist; it. The object has been renamed.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. Object name *object_name* in statement type *statement_type* does not exist; it has been renamed.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3016E An error occurred while processing the *obj_name* in the statement

type of *stmt_typ*. The object *obj_name2* already exists.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3017E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The object *obj_name2* does not exist.**

Explanation:
An attempt was made to drop a clone table, but the specified base table does not have a clone table, or the clone table has been dropped. The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a WSL, correct the SQL statement in the work statement list (WSL) and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3018E **An error occurred while processing the *object-name* object in the statement type of *statement-type*. The column *column-name* is nullable. PRIMARY KEY, HASH KEY or UNIQUE constraint cannot be defined on it.**

Explanation

A nullable column cannot be used to define a primary key, hash key, or unique constraint.

object-name
The name of the object.

statement-type
The type of statement.

column-name
The name of the nullable column that caused the error.

System action:
Processing stops.

User response:
Specify a valid column for the primary key, hash key, or unique constraint.

ADB3020W **The *obj_name* object in the statement type of *stmt_type* and that are referred in CREATE, ALTER, COMMENT, DROP, EXCHANGE, LABEL, or RENAME statements might not exist during NSP run time.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3021E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. *obj_name2* is not registered in the XML Schema Repository.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3022E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The**

object is a history table and cannot be explicitly dropped.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action:
After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3023E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action:
After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3024E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action:
After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3025E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The column *obj_name2* already exists in the table.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action:
After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3026E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The column *obj_name2* does not exist in the table or is defined as a NOT NULL column.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action:
After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3027E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The EXCLUDE NULL KEYS clause is ignored with UNIQUE indexes.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action:
After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response:

If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3028E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The EXCLUDE NULL KEYS clause cannot be specified if a BUSINESS_TIME WITHOUT OVERLAPS index is also specified.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action:
After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3029E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The EXCLUDE NULL KEYS clause cannot be specified if the index is defined with an XML-index-specification.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate Report to indicate an error with the identified SQL statement.

System action:
After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3030E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The EXCLUDE NULL KEYS clause cannot be specified if the index is defined with a key-expression.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3031E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The EXCLUDE NULL KEYS clause cannot be specified if the index is defined with an INCLUDE (column name) clause.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3032E **An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. The EXCLUDE NULL KEYS clause cannot be specified if the index is defined as a partitioning index.**

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object

Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3033E An error occurred while processing the *obj_name* object in the statement type of *stmt_typ*. Add column *obj_name2*. The requested operation or usage does not apply to the created global temporary table.

Explanation:
The syntax of the generated SQL statements is being validated to check consistency. This message is written to the Validate report to indicate an error with the identified SQL statement.

System action:
After the Validate report is generated, a return code of 8 is set, and processing stops.

User response:
If you are validating a work statement list (WSL), correct the SQL statement in the WSL and re-validate the WSL. If you are comparing objects with DB2 Object Comparison Tool, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3034E An error occurred while processing the *<object_name> <object_type>* in the *<statement_type>* statement. The *<object_type> <object_name>* is already archive enabled or the wrong type of table is specified to be archive enabled.

Explanation:
The SQL statement referred to in this message specifies an archive table name that is already archive enabled or specifies a table cannot be specified as archive enabled. This error message is written to the Validate Report to indicate an error with the identified SQL statement.

User response:
Verify that the correct table is specified. Then, if you are validating a WSL, correct the SQL statement in the work statement list (WSL) and re-validate the WSL.
If you are using DB2 Object Comparison to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3035E An error occurred while processing the *<object_name> <object_type>* in the *<statement_type>* statement. The *<object_type> <object_name>* is not archive enabled.

Explanation:
The SQL statement referred to in this message specifies an archive table name that is not archive enabled. This message is written to the Validate Report to indicate an error with the identified SQL statement.

User response:
Verify that the correct table is specified. Then, if you are validating a WSL, correct the SQL statement in the work statement list (WSL) and re-validate the WSL.
If you are using DB2 Object Comparison to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3036E An error occurred while processing *<stmt_type> <obj_type>* statement: *<err_msg>*

Explanation:
The SQL statement referred to in this message is invalid because of the specified reason. This message is written to the VALOUT data set to indicate an error with the identified SQL statement.

System action:
After the Validate Report is generated, a return code of 8 is set, and processing stops.

User response:
Correct the SQL statement in the work statement list(WSL) and re-validate the WSL. If you are using DB2 Object Comparison to compare objects, correct the SQL statement in the DDL source and re-generate the compare job.

ADB3037E The ADB2IRXCA function *<stmt_type> <obj_type>* failed with the following error: *<err_msg>*

Explanation:
The ADB2IRXCA function referred to in this message failed with the specified message.

System action:
Processing stops.

ADB3101E Unexpected sqlcode in *error_function*.

System action:
No system action is taken.

User response:
Fix the problem and try again

ADB3201E Applying the DBNAME *obj_name1* mask results in the creation of an implicit or system-reserved database, *obj_name2*.

Explanation:
The specified DBNAME mask definition results in the creation of an implicit or system-reserved database, which is not valid because the database is not accepted by DB2 *obj_name1* and *obj_name2*.

System action:

A return code of 8 is set and processing stops.

User response:
Correct the definition of the DBNAME mask, and resubmit the job.

ADB3202W The data set name *obj_name1*that is referred to in an UNLOAD statement might not exist after masks are applied.

System action:
Processing continues.

User response:
Evaluate the masks that you are using to determine their effect on the specified data set. If the data set does not exist after the masks are applied, correct the problem and resubmit the job.

ADB3301E The overwrite value for HASHSPC must be numeric followed by character K, M, or G. Overwrite Value = *text1*.

Explanation:
The use of masking was specified, and the value that is specified for HASHSPC is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask. If a specific value is specified for HASHSPC, ensure that the value is an integer value that is followed by the character K, M, or G. If a REXX user exit is specified for HASHSPC, ensure that the REXX user exit is coded so that it returns an integer value followed with the character K, M, or G. After the corrections are made, resubmit the job.

ADB3302E The overwrite value for TBINLOBL must be numeric and in a valid range. Overwrite Value = *text1*.

Explanation:
The use of masking was specified, and the value that is specified for TBINLOBL is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask. If a specific value is specified for TBINLOBL, ensure that the value is an integer value. If a REXX user exit is specified for TBINLOBL, ensure that the REXX user exit is coded so that it returns an integer value. After the corrections are made, resubmit the job.

ADB3303E The overwrite value for DTINLOBL must be numeric and in a valid range. Overwrite Value = *text1*.

Explanation:
The use of masking was specified, and the value that is specified for DTINLOBL is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask. If a specific value is specified for DTINLOBL, ensure that the value is an integer value. If a REXX user exit is specified for DTINLOBL, ensure that the REXX user exit is coded so that it returns an integer value. After the corrections are made, resubmit the job.

ADB3304E The overwrite value for TBINLOBL exceeded the maximum length of a column. Overwrite Value = *text1*

Explanation:
The use of masking was specified, and the value that is specified for TBINLOBL is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask. If a specific value is specified for TBINLOBL, ensure that the overwrite value does not exceed the maximum length of a column. If a REXX user exit is specified for TBINLOBL, ensure that the REXX user exit is coded so that it returns an overwrite value that will not exceed the maximum length of a column. After the corrections are made, resubmit the job.

ADB3305E The overwrite value for DTINLOBL exceeded the maximum length of a distinct type.

Explanation:
The use of masking was specified, and the value that is specified for DTINLOBL is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask. If a specific value is specified for DTINLOBL, ensure that the overwrite value does not exceed the maximum length of a distinct type. If a REXX user exit is specified for DTINLOBL, ensure that the REXX user exit is coded so that it returns an overwrite value that will not exceed the maximum length of a distinct type. After the corrections are made, resubmit the job.

ADB3306E The overwrite value for the HASHSPC mask is not within the valid range. Overwrite Value = *text1*.

Explanation:
The use of masking was specified, and the value that is specified for HASHSPC is not within the valid range.

System action:
Processing stops.

User response:
If a REXX user exit is specified for the HASHSPC mask, ensure that the REXX user exit is coded so that it returns an overwrite value that is in the valid range. After the corrections are made, resubmit the job.

ADB3307E **The character that is specified in the SINGLECH mask is equivalent to a wildcard (*) character. Single character = *text1*.**

Explanation:
The use of masking was specified, but the character that is specified for the SINGLECH mask is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask. After the corrections are made, resubmit the job.

ADB3308E **The character that is specified in the SINGLECH mask is invalid. Single character = *text1*.**

Explanation:
The use of masking was specified, but the character that is specified for the SINGLECH mask is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask. After the corrections are made, resubmit the job.

ADB3309E **The escape character that is specified in the SINGLECH mask is equivalent to a wildcard (*) character or to the specified single character. Escape character = *text1*.**

Explanation:
The use of masking was specified, but the escape character that is specified for the SINGLECH mask is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask. After the corrections are made, resubmit the job.

ADB3310E **The escape character that is specified in the SINGLECH mask is invalid. Escape character = *text1*.**

Explanation:
The use of masking was specified, but the escape character that is specified for the SINGLECH mask is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask. After the corrections are made, resubmit the job.

ADB3311E **The overwrite value for the AUDIT mask is not valid. Overwrite Value = *text1*.**

Explanation:
The use of masking was specified, but the value that is specified for the AUDIT mask is not valid.

System action:
Processing stops.

User response:
If a REXX user exit is specified for the AUDIT mask, ensure that the REXX user exit is coded so that it returns an overwrite value of ALL, CHANGES or NONE.

ADB3312E **The overwrite value for the CLOSE, TSCLOSE or IXCLOSE mask is invalid. Overwrite Value = *text1*.**

Explanation:
The use of masking was specified, but the value that is specified for CLOSE, TSCLOSE or IXCLOSE is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask and resubmit the job. If a REXX user exit is specified for the mask, ensure that the REXX user exit is coded so that it returns an overwrite value of YES or NO.

ADB3313E **The overwrite value for the CCSID mask is invalid. Overwrite Value = *text1*.**

Explanation:
The use of masking was specified, but the value that is specified for the CCSID mask is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask and resubmit the job. If a REXX user exit is specified for the CCSID mask, ensure that the REXX user exit is coded so that it returns an overwrite value of ASCII, EBCDIC or UNICODE.

ADB3314E **The mask value for the SYNSCHEMA mask is too long. Overwrite Value = *text1*.**

Explanation:
The use of masking was specified, but the value that is specified for the SYNSCHEMA mask is too long. The maximum length is 128 characters.

System action:
Processing stops.

User response:
Correct the definition of the mask and resubmit the job. If a REXX user exit is specified for the SYNSCHEMA mask, ensure that the REXX user exit is coded so that it returns an overwrite value in the valid range.

ADB3315E The mask type does not support object-specific masking. Mask type = *text1*.

Explanation:
Some mask types are not supported for object-specific masking because they either are too general to determine the objects in question, or they do not refer to objects.

System action:
Processing stops.

User response:
Correct the definition of the mask. Change the mask to be non-object-specific, or change the mask type to a more specific mask type. For example, use TBNAME instead of NAME if masking a specific table object. After the corrections are made, regenerate, and then resubmit the job.

ADB3316E The object specification of an object-specific mask does not match the format that is required for the object that is being masked by the mask type. Mask type = *text1*.

Explanation:
Mask types require either a single qualifier specification or a qualifier and a name specification depending on the object that is being masked.

System action:
Processing stops.

User response:
Correct the definition of the mask. Change the object specification to match the required specification. For example, TBNAME:TBSCH1.TBNAME:TBNAME,NEWTB requires both TBSCH1 and TBNAME in the object specification. After the corrections are made, regenerate, and then resubmit the job.

ADB3317W The external name of a Java program cannot be masked due to the length of the name.

Explanation:
Java external names that are greater than 128 characters cannot be masked.

System action:
Processing continues.

User response:
Change the Java external name manually.

ADB3318W *text1* could not convert characters from CCSID(*text2*) to CCSID(37).

Explanation:
The program could not convert the characters to CCSID(37).

System action:
Processing continues.

User response:
Use a valid CCSID mask value. See the DB2 for z/OS SQL Reference for valid values.

ADB3319W The mask value for DSSIZE on the table space *text1* was skipped because the table space is type *text2*.

Explanation:
The attribute DSSIZE is only valid in a partitioned table space, partition-by-growth table space, range-partitioned universal table space, and LOB table space.

System action:
Processing continues. No system action is taken.

User response:
None.

ADB3320W SEGSIZE was masked from 0 to *text1* for table space *text2*. The value might change the table space type.

Explanation:
If the original setting for SEGSIZE mask was 0, then the input mask value might change the table space type. For example, specifying the SEGSIZE mask might convert a partitioned table space to a range-partitioned universal table space (UTS). If a table in a UTS has a partitioned index and the partitioned index needs to be created, DB2 might generate a SQLCODE=-662 error during execution.

System action:
Processing continues.

User response:
If necessary, specify a valid input mask value, regenerate, and resubmit the job.

ADB3321E The mask name is too long after applying renames from Name = *<old name>* to Newname = *<new name>*.

Explanation:
The use of masking or renames is specified. The value that is specified for masking or renames is too long.

System action:
Processing stops.

User response:
Correct the name that is defined for the mask or renames, and try again. If a REXX user exit is specified for masks, ensure that the REXX user exit is coded so that a value in the valid range is returned. After the corrections are made, regenerate, and resubmit the job.

ADB3322E **The overwrite value for the TRACKMOD is invalid. Overwrite Value = *text1***

Explanation:
The use of masking or renames was specified, but the value that is specified for the TRACKMOD mask is not valid.

System action:
Processing stops.

User response:
Correct the definition of the TRACKMOD mask. If a REXX user exit is specified for the TRACKMOD mask, ensure that the REXX user exit is coded so that it returns an overwrite value that is either YES or NO. After the corrections are made, regenerate, and resubmit the job.

ADB3323E **The overwrite value for the DCAPTURE (DATA CAPTURE) mask is not valid. Overwrite Value = *text1***

Explanation:
The use of masking or renames is specified, but the value that is specified for the DCAPTURE mask is not valid.

System action:
Processing stops.

User response:
Correct the definition of the DCAPTURE mask. If a REXX user exit is specified for DATA CAPTURE, ensure that the REXX user exit is coded so that it returns an overwrite value of NONE or CHANGES. After the corrections are made, regenerate, and resubmit the job.

ADB3324E **The overwrite value for *text1* FREEPAGE is not correct and must be numeric in the range of 0 - 255. Overwrite Value = *text2*.**

Explanation:
The use of masking was specified, but the value that is specified for the FREEPAGE attribute overwrites FREEPG or TSFREEPG or IXFREEPG is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of the FREEPG or TSFREEPG or IXFREEPG overwrites and try again. If a specific value is specified for FREEPG or TSFREEPG or IXFREEPG overwrites, ensure that the value is an integer value in the range of 0 - 255. If a REXX user exit is specified for FREEPG or TSFREEPG or IXFREEPG overwrites, ensure that the REXX user exit is coded so that it returns an integer overwrite value in the range of 0 - 255. After the corrections are made, resubmit the job.

ADB3325E **The overwrite value for *text1* is not correct and must be numeric in the range of 0 - 99.**

Explanation:
The use of masking was specified, and the value that is specified for PCTFREE attribute overwrites PCTFREE or TSPCTFREE or IXPCTFREE is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of the PCTFREE or TSPCTFREE or IXPCTFREE overwrites and try again. If a specific value is specified for PCTFREEE or TSPCTFREE or IXPCTFREE overwrites, ensure that the value is an integer value in the range of 0 - 99. If a REXX user exit is specified for PCTFREE or TSPCTFREE or IXPCTFREE overwrites, ensure that the REXX user exit is coded so that it returns an integer overwrite value in the range of 0 - 99. After the corrections are made, resubmit the job.

ADB3326E **The overwrite value for *text1* is not correct and must be numeric in the range of 0-2147483647 or SYSTEM. Overwrite Value = *text2*.**

Explanation:
The use of masking was specified, and the value that is specified for LOCKMAX is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of LOCKMAX overwrite and try again. If a specific value is specified for LOCKMAX, ensure that the value is an integer value in the range of 0 - 2147483647 or SYSTEM. If a REXX user exit is specified for LOCKMAX, ensure that the REXX user exit is coded so that it returns an integer overwrite value in the range of 0 - 2147483647 or SYSTEM. After the corrections are made, resubmit the job.

ADB3327E The overwrite value for *text1* is not correct and should be either YES or NO. Overwrite Value = *text2*

Explanation:
The use of masking was specified, and the value that is specified for ERASE attribute overwrites ERASE or TSERASE or IXERASE is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of the ERASE or TSERASE or IXERASE overwrites and try again. If a specific value is specified for ERASE or TSERASE or IXERASE overwrites, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for ERASE, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

ADB3328E The specified authorization ID, *authorization_id*, is not valid.

Explanation:
The authorization ID for the *authorization_id* or host variable in the SQL SET CURRENT SQLID statement is not your primary authorization ID or one of the associated secondary authorization IDs.

System action:
The SET CURRENT SQLID statement cannot be executed. The current SQL ID is not changed..

User response:
Correct the error in the statement or contact the security administrator to have the authorization ID defined for your use.

ADB3329E The inmask ends or outmask starts with a comma for field *>masktype<*.

Explanation:
The inmask value ends with a comma for MASK field *>masktype<* or the outmask value starts with a comma for MASK field *>masktype<*.

System action:
Processing stops.

User response:
Remove the comma.

ADB3331E The overwrite value for LOGGED is not correct and should be either YES or NO.

Explanation:
The use of masking was specified, and the value that is specified for LOGGED attribute overwrites is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of the LOGGED overwrites and try again. If a specific value is specified for LOGGED overwrites, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for LOGGED overwrites, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

ADB3332E The overwrite value for GBPCACH is not valid.

Explanation:
The use of masking was specified, and the value that is specified for GBPCACH attribute overwrites GBPCACH or TSGBPCACH or IXGBPCACH is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of the GBPCACH or TSGBPCACH or IXGBPCACH overwrites and try again. If a specific value is specified for GBPCACH or TSGBPCACH or IXGBPCACH overwrites, ensure that the overwrite value is SYSTEM, CHANGED, ALL, or NONE. If a REXX user exit is specified for GBPCACH or TSGBPCACH or IXGBPCACH overwrites, ensure that the REXX user exit is coded so that it returns an overwrite value that is SYSTEM, CHANGED, ALL, or NONE. After the corrections are made, resubmit the job.

ADB3333E The overwrite value for APPEND is not correct and should be either YES or NO.

Explanation:
The use of masking was specified, and the value that is specified for the APPEND attribute overwrite is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of the APPEND overwrite and try again. If a specific value is specified for the APPEND overwrite, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for APPEND overwrite, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

ADB3334E The overwrite value for TSPARTS is not correct and must be numeric in the range of 0-4096.

Explanation:
The use of masking was specified, and the value that is specified for TSPARTS is not valid.

System action:

Processing is discontinued with return code 12.

User response:
Correct the definition of the TSPARTS overwrite and try again. If a specific value is specified for TSPARTS, ensure that the overwrite value is an integer value in the range of 0-4096. If a REXX user exit is specified for TSPARTS, ensure that the REXX user exit is coded so that it returns an integer overwrite value in the range of 0 - 4096. After the corrections are made, resubmit the job.

ADB3335E **The overwrite value for the LOCKSIZE mask is not valid.**

Explanation:
The use of masking was specified, but the value that is specified for the LOCKSIZE mask is not valid.

System action:
Processing stops.

User response:
Correct the definition of the mask and resubmit the job. If a REXX user exit is specified for the LOCKSIZE mask, ensure that the REXX user exit is coded so that it returns an overwrite value of TABLE, TABLESPACE, LOB, PAGE, ROW, or ANY.

ADB3336E **The overwrite value for PADDED is not correct and should be either YES or NO.**

Explanation:
The use of masking was specified, but the value that is specified for the PADDED attribute overwrite is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of the PADDED overwrite and try again. If a specific value is specified for the PADDED overwrite, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for the PADDED overwrite, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

ADB3337E **The overwrite value for MAXROWS is not correct and must be numeric in the range of 0-255.**

Explanation:
The use of masking was specified, but the value that is specified for MAXROWS is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of the MAXROWS overwrite and try again. If a specific value is specified for the MAXROWS overwrite, ensure that the overwrite value is an integer value in the range of 0-255. If a REXX user exit is specified for MAXROWS, ensure that the REXX user exit is coded so that it returns an overwrite value in the range of 0-255. After the corrections are made, resubmit the job.

ADB3338E **The overwrite value for VOLATILE is not correct and should be either YES or NO.**

Explanation:
The use of masking was specified, but the value that is specified for VOLATILE is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of the VOLATILE overwrite and try again. If a specific value is specified for the VOLATILE overwrite, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for VOLATILE, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

ADB3339E **The overwrite value for COPY is not correct and should be either YES or NO.**

Explanation:
The use of masking was specified, but the value that is specified for COPY is not valid.

System action:
Processing is discontinued with return code 12.

User response:
Correct the definition of the COPY overwrite and try again. If a specific value is specified for the COPY overwrite, ensure that the overwrite value is YES or NO. If a REXX user exit is specified for COPY, ensure that the REXX user exit is coded so that it returns an overwrite value that is YES or NO. After the corrections are made, resubmit the job.

ADB3340E **The VER overwrite mask syntax is missing a comma after the word VER.**

Explanation:
The use of VER masking was specified, but a comma was not present between the VER string and the overwrite mask.

System action:
Processing is discontinued with return code 12.

User response:
Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3341E The VER overwrite mask syntax has an invalid operand. Operand *text1* mask type was *text2*.

Explanation:
The use of VER masking was specified, but an invalid operand was given. The valid operands are EQ, NE, GT, LT, RANGE and LIST.

System action:
Processing is discontinued with return code 12.

User response:
Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3342E The VER overwrite mask syntax is missing an equal sign after the word RC.

Explanation:
The use of VER masking was specified, but an equal sign was not present after the RC string.

System action:
Processing is discontinued with return code 12.

User response:
Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3343E The VER overwrite mask syntax is missing the word RC.

Explanation:
The use of VER masking was specified, but the RC string was not present.

System action:
Processing is discontinued with return code 12.

User response:
Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3344E The VER overwrite mask syntax is for range operation, but two values are not present after the RANGE string.

Explanation:
The use of VER masking was specified, but two values are missing after the RANGE string.

System action:
Processing is discontinued with return code 12.

User response:
Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3345E The VER overwrite mask syntax is for range operation, but more than two values are present after the RANGE string.

Explanation:
The use of VER masking was specified, but there are too many values after the RANGE string.

System action:
Processing is discontinued with return code 12.

User response:
Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3346E RC = *return_code*. Error processing *mask_name*. Value is *attr_value*. Verification failed for operation *verify_oper*.

Explanation:
The attribute value does not conform to the verification mask.

System action:
Processing is discontinued with return code 8 or 12.

User response:
Correct the attribute value and try again. After the corrections are made, resubmit the job.

ADB3348E The VER overwrite mask syntax has an invalid operand for mask *maskname*. The operand is *veroper*.

Explanation:
The use of VER masking was specified, but an invalid operand was given. Some operands are not allowed for some masks.

System action:
Processing is discontinued with return code 12.

User response:
Correct the syntax of the VER mask and try again. After the corrections are made, resubmit the job.

ADB3356W To migrate DB2V8 masking input, OWNER, TBOWNER and IXOWNER will mask both owner and schema fields. Masking function will duplicate OWNER, TBOWNER and IXOWNER masks to respective SCHEMA, TBSCHEMA and IXSCHEMA masks in migrating DB2V8 masking input to help users not to change their masking datasets. In the future, this mapping will be removed and users need to consider changing their masking datasets, if they have any masks OWNER, TBOWNER and IXOWNER defined in mask datasets prior to DB2 V9.

Explanation

If you are using DB2 9 for z/OS or later, the OWNER, TBOWNER, and IXOWNER masks also apply to the corresponding schema masks. Specifically:

- If you define an OWNER mask, a SCHEMA mask is implicitly defined with the same values.
- If you define a TBOWNER mask, a TBSCHEMA mask is implicitly defined with the same values.
- If you defined an IXOWNER mask, an IXSCHEMA mask is implicitly defined with the same values.

This reason for this behavior is a change in the Db2 catalog. Beginning in DB2 9 for z/OS, the table and index creator columns in the catalog changed from storing the authorization ID of the owner to storing the schema. To save you from having to change your mask definitions to account for these column changes, Db2 Admin Tool began implicitly creating the corresponding schema masks for OWNER, TBOWNER, and IXOWNER masks. As a result, these owner masks still apply to the table and index creator fields.

In the future, these implicit schema mask definitions will be removed.

System action:
Processing continues.

User response:
If you have OWNER, TBOWNER, and IXOWNER masks defined, consider changing your mask definitions to SCHEMA, TBSCHEMA, and IXSCHEMA.

ADB3357E **The overwrite value for INSALGO is not correct and must be numeric in the range of 0 - 2. Overwrite Value = *value***

Explanation

This job uses masking. However, the overwrite value that was specified for the INSALGO mask is not valid.

value
The specified overwrite value.

System action:
Processing stops.

User response

Correct the definition of the INSALGO mask. Ensure that the overwrite value that is specified is the integer value 0, 1, or 2. If you specified a REXX user exit instead of an explicit value, ensure that the user exit is coded to return one of those integer values.

The following example shows a valid INSALGO mask definition:

```
INSALGO:DB1.TS1,2
```

After you correct the mask definition, resubmit the job.

Related concepts
“Mask definitions” on page 274
Mask definitions are reusable; you can define a mask once and use it repeatedly.

ADB3358W **To migrate DB2 V8 masking input, the masking function previously duplicated OWNER, TBOWNER, and IXOWNER masks to respective SCHEMA, TBSCHEMA and IXSCHEMA masks.**

Explanation

You specified at least one OWNER, TBOWNER, or IXOWNER mask. Be aware that Db2 Admin Tool no longer implicitly creates corresponding schema masks for these owner masks. This change avoids unintended schema masking and gives you more control over the owner and schema attributes that you want to mask.

Beginning in DB2 9 for z/OS, the table and index creator columns in the catalog changed from storing the authorization ID of the owner to storing the schema. To prevent you from having to change your mask definitions to account for these column changes, Db2 Admin Tool began implicitly creating the corresponding schema masks for OWNER, TBOWNER, and IXOWNER masks if you were using DB2 9 or later. Specifically:

- If you defined an OWNER mask, a SCHEMA mask was implicitly defined with the same values.
- If you defined a TBOWNER mask, a TBSCHEMA mask was implicitly defined with the same values.
- If you defined an IXOWNER mask, an IXSCHEMA mask was implicitly defined with the same values.

These implicit masks are no longer defined.

System action:
Processing continues.

User response:
Consider updating your masking data sets if they contain any OWNER, TBOWNER, or IXOWNER masks. If you defined those masks before DB2 9, consider whether they need to be changed to SCHEMA, TBSCHEMA, or IXSCHEMA masks.

Related concepts
“Mask definitions” on page 274
Mask definitions are reusable; you can define a mask once and use it repeatedly.

ADB5000E **An invalid value specified for parameter *insert1*.**

Explanation:
An invalid value was specified for the parameter.

System action:
Processing stops.

User response:
Specify a valid value for the parameter.

ADB5001E The PLAN= parameter was not found.

Explanation:
The **ADBOPT** parameter of **PLAN=** is required for ADBTEPA.

System action:
Processing stops.

User response:
Provide the **PLAN=** parameter in the ADBOPT DD card.

ADB5002E The ADBTEPA invocation was not from an APF-authorized environment.

System action:
Processing stops.

User response:
Use APF to authorize all data sets in the STEPLIB.

ADB5003E A failure occurred attempting command *RexxCmd1*.

Explanation:
The provided command failed for an undetermined reason.

System action:
Processing stops.

User response:
If possible, resolve the problem and run the *RexxCmd1* command again.

ADB5005E A DB2 pending change will be lost by dropping the object.

Explanation:
The input statement was a DROP, the object involved had a DB2 pending change which would be lost, and the PENDINGCHANGESCHECK='YES' parameter was specified.

System action:
Processing stops.

User response:
None.

ADB5007E An invalid input parameter *InvalidParm* was encountered.

Explanation:
The parameter is unrecognized.

System action:
Processing stops.

User response:
Remove the unrecognized parameter.

ADB5008E Either the SSID= or PLAN= parameter was not provided.

System action:
Processing stops.

User response

1. Provide the **PLAN=***plan* setting in the ADBOPT DD card.
2. Provide the **SSID()** parameter as a parameter to the program.

ADB5009E A non-zero SQL code was issued.

System action:
Processing stops.

User response:
Investigate the specific SQL code and take remedial action.

ADB5010W The DB2 Version could not be determined.

Explanation:
This message indicates a positive, non-zero return code from the SQL CONNECT statement.

System action:
The product assumes a DB2 Version 6 level, and processing continues.

User response:
None.

ADB5011W ADBCHKPT update failed for WORKLIST(*Wklist*) during RESTART(NO) processing.

System action:
Processing continues.

User response:
None.

ADB5017E The ADBCHKPT checkpoint table does not exist.

System action:
Processing stops.

User response:
Check the package qualifier of ADBTEP2.

ADB5021W The preceding query was cancelled by RLF after successful retrieval of *RecCnt* rows.

System action:
The cursor is closed and processing continues.

User response:
None.

ADB5025E A "Not Found" condition was encountered during an open.

System action:
Processing continues.

User response:
None.

ADB5028E An authorization error occurred during -START.

System action:
Processing stops.

User response:
Grant the job submitter ID the necessary authority and restart the batch statement list.

ADB5029E An error occurred during -START.

Explanation:
An unrecognized error occurred while attempting the **-START** command.

System action:
Processing stops.

User response:
Examine the output and take remedial action.

ADB5031W The following list of *n* tables(s) were not enabled for *operation* on the accelerator *accelerator-name* prior to the recreate operation: *table-names*

Explanation

The listed tables were not enabled for the specified operation, because they were not enabled for this operation before the recreate.

n
The number of tables that were not enabled for the operation.

operation
The operation. Possible values are acceleration or replication.

accelerator-name
The name of the accelerator server.

table-names
The qualified names of the tables that were not enabled for the operation. Each table name is separated by a comma.

System action:
Processing continues.

User response

No action is required.

ADB5034E Delete failed for ADBCHKPT control record for WORKLIST(*WorkList*).

System action:
Processing stops.

User response:
Resubmit the job to complete processing.

ADB5035E Invalid input parm term character.

System action:
Processing stops.

User response:
Specify a valid term character.

ADB5036E A trailing paren has been omitted or no value was provided.

Explanation:
A trailing parenthesis was omitted or no value was provided.

System action:
Processing stops.

User response:
Specify a trailing parenthesis or provide a value.

ADB5037E Error in MAXE input parameter paren

Explanation:
An error in the MAXE input parameter parenthesis occurred.

System action:
Processing stops.

User response:
Provide a valid MAXE parameter.

ADB5043E Restart processing was halted due to a command mismatch.

Explanation:
The command from the last run does not match the command from the restarted run.

System action:
Processing stops.

User response:
Verify that the statement being restarted has not been changed. Alternatively, you can start the job run with the parameter RESTART(FORCE). ADBTEP2 will skip the changed command and continue the run.

ADB5044I * INPUT STATEMENT *input-statement***

Explanation

The specified statement was submitted.

input-statement

The submitted statement, such as an SQL statement, a Db2 utility statement, or other command.

System action:
Processing continues.

User response

No action is required.

ADB5045I ***** INPUT ASSOCIATED WITH A FOLLOWING STMT:**

Explanation

This line identifies input, such as a data set template, that is needed for a statement that is to be run by ADBTEP2. The statements that are to be run are identified by message ADB5047I in the ADBTEP2 report .

Because **Restart Report Only** was specified, this report is a simulation only. No actual statements will be executed.

System action:
Processing continues.

User response

No action is required.

ADB5046I ***** INPUT STATEMENT SKIPPED:**

Explanation:
ADBTEP2 skipped the listed statement. Because ADBTEP2 was restarted without **Report Only** and a checkpoint exists, any statements before the checkpoint are skipped.

System action:
Processing continues.

User response

No action is required.

ADB5047I ***** INPUT STATEMENT WILL BE RUN FOR A NORMAL RESTART:**

Explanation

The listed statement will be run by ADBTEP2 during a restart.

Because **Restart report only** was specified, this report is a simulation only. No actual statements will be executed.

System action:
Processing continues.

User response

No action is required.

ADB5050W **The above statement was not approved for authorization switching.**

Explanation:
The statement preceding this warning message is not approved for authorization switching.

System action:
Processing continues.

User response

No action is required.

Related concepts

“Authorization switching” on page 893

Authorization switching is a facility within Db2 Admin Tool that is used to execute DDL and DCL under the authority of another user. The facility does not cover other statement types, including Db2 utility statements and DSN subcommands such as FREE PACKAGE and BIND PLAN.

ADB5051E **An error occurred in the `CHANGEID()` input parameter**

System action:
Processing stops.

User response:
Provide the correct **`CHANGEID()`** parameter and value.

ADB5052E **The CONNECT statement contained syntax errors.**

System action:
Processing continues.

User response:
Specify a valid CONNECT statement.

ADB5054E **The SET CONNECTION statement contained syntax errors.**

System action:
Processing continues.

User response:
Specify a valid SET CONNECTION statement.

ADB5056E **There is an error in the `CHANGEID()` input parameter value.**

System action:
Processing stops.

User response:
Provide the correct `CHANGEID()` parameter and value.

ADB5057E The SET QUERYNO statement contained syntax errors.

System action:
Processing continues.

User response:
Specify a valid SET QUERYNO statement.

ADB5058E The RELEASE statement contained syntax errors.

System action:
Processing continues.

User response:
Specify a valid RELEASE statement.

ADB5060I * Implicit commit performed, number: *number-commits***

Explanation

The specified number of implicit commit operations were performed. (In all Db2 environments, the normal termination of a process is an implicit commit operation.)

number-commits
The number of implicit commit operations.

System action:
Processing continues.

User response

No action is required.

ADB5063E The ADBCHKPT control record for WORKLIST(*WorkList*) is missing.

System action:
Processing stops.

User response:
Provide the WORKLIST(*WorkList*) parameter and value.

ADB5064E There is an SQL buffer overflow. The maximum size is *Maxsize*.

System action:
Processing stops.

User response:
Specify a larger region size.

ADB5067E The command Command is not supported, or execs are not in SYSEXEC/SYSPROC.

System action:
Processing stops.

User response:
Provide a SYSEXEC DD card.

ADB5071E The ADBPART table does not exist.

System action:
Processing stops.

User response:
Check the qualifier of package ADBTEP2.

ADB5073W Keys do not match for part *PartNo* .

Explanation:
Limitkeys do not match between unload and load. Processing of data might proceed serially.

System action:
Processing continues.

User response:
None.

ADB5074W Unloads will be performed using DB2.

Explanation:
When a condition is encountered which requires a DB2 unload, the unload will be performed by DB2, not by HPU.

System action:
Processing continues.

User response:
None.

ADB5075I * CONDITIONAL INPUT STATEMENT:**

Explanation:
The listed input statement to ADBTEP2 is conditional. To determine whether this condition is currently evaluated as true or false, see message ADB5076.

System action:
Processing continues.

User response

No action is required.

ADB5076I * CONDITIONAL INPUT STATEMENT IS EVALUATED AS *boolean_value***

Explanation

This message reports whether the conditional statement in ADB5075I is true or false.

boolean_value
TRUE or FALSE

System action:
Processing continues.

User response

No action is required.

ADB5077I ***** INPUT STATEMENT BYPASSED DUE TO FALSE CONDITION:**

Explanation:
The listed statement is skipped, because the conditional statement in ADB5075I is false.

System action:
Processing continues.

User response

No action is required.

ADB5080E **A restart with a different unload method is not allowed.**

Explanation:
It is not permitted to change the UNLOAD method on restart.

System action:
Processing continues.

User response:
Either resubmit the restart with DB2 (parm **`UNLOAD(HPU)`** or start the run from the beginning **`RESTART(NO)`**.

ADB5081E **A restart with a different unload method is not allowed.**

Explanation:
It is not permitted to change the UNLOAD method on restart.

System action:
Processing continues.

User response:
Either resubmit the restart with DB2 (parm **`UNLOAD(DB2)`** or start the run from the beginning **`RESTART(NO)`**.

ADB5094E **The held DSN commands have been queued on SYSIN and will be retried.**

System action:
Processing continues.

User response:
None.

ADB5100E **No restart was requested and no checkpoint was found. This was an abnormal run, and cannot be restarted.**

System action:
Processing stops..

User response:
None.

ADB5105E **The command Command is not supported or the execs are not in SYSEXEC/SYSPROC.**

System action:
Processing stops.

User response:
Provide a SYSEXEC DD containing the product execs.

ADB5106I **The following error is tolerated. The value of the parameter MAXERRORS determines the number of errors that are tolerated.**

Explanation:
An error occurred but processing continues because the MAXERROR parameter is specified with a value of -1 or a value between 1 and 99.

System action:
Processing continues.

User response:
If you do not want error tolerance, set the MAXERRORS parameter to 0. Specify a value of -1 to indicate that the program should tolerate an unlimited number of errors for DSN commands. Specify a value between 1 and 99 to indicate the number of errors that the program should tolerate.

ADB5118I **Request to switch to alternative userid, *auth-switch-id*, has been granted. This ID will be used for the SQL statement that is validated. Non-validated SQL will run under the submitter's authority.**

Explanation

The request to switch to an alternative user ID was granted. The named ID will be used for the SQL statement that is validated.

Non-validated SQL, which will run under the submitter's authority, includes the following statements that are not eligible for authorization switching:

- A DDL statement that is not one of the following statements: ALTER (TABLE, MASK, PERMISSION, FUNCTION, and TRIGGER), COMMENT, LABEL, CREATE, SET, or GRANT (except system privilege).
- Any DDL that was manually added to the file or edited.
- DDL that was not run within 8 days of being created.

If ineligible statements are encountered, Db2 Admin Tool switches from the requesting auth-switch ID to the submitter ID. Then, when an eligible statement is encountered, Db2 Admin Tool switches back to the auth-switch ID .

auth-switch-id
The ID that will be used for the SQL statement that is validated.

System action:
Processing continues.

User response

No action is required.

Related concepts

"Authorization switching" on page 893
Authorization switching is a facility within Db2 Admin Tool that is used to execute DDL and DCL under the authority of another user. The facility does not cover other statement types, including Db2 utility statements and DSN subcommands such as FREE PACKAGE and BIND PLAN.

ADB5121E Command failed: *command*

Explanation:
The listed Db2 command failed.

System action:
Processing stops.

User response:
If possible, address the problem and restart.

ADB5129W Some statements that requested to be auth-switched were not validated and were attempted or ran under the submitter's authority. Message ADB5050W identifies those statements.

Explanation

Authorization switching was requested; however the job included at least one statement that is not eligible for authorization switching. When these ineligible statements are encountered, Db2 Admin Tool switches from the requesting auth-switch ID to the submitter ID. Then, when an eligible statement is encountered, Db2 Admin Tool switches back to the auth-switch ID .

Ineligible statements can be one of the following statements:

- A DDL statement that is not one of the following statements: ALTER (TABLE, MASK, PERMISSION, FUNCTION, and TRIGGER), COMMENT, LABEL, CREATE, SET, or GRANT (except system privilege).
- Any DDL that was manually added to the file or edited.
- DDL that was not run within 8 days of being created.

System action:
Processing continues.

User response

No action is required.

Related concepts

"Authorization switching" on page 893
Authorization switching is a facility within Db2 Admin Tool that is used to execute DDL and DCL under the authority of another user. The facility does not cover other statement types, including Db2 utility statements and DSN subcommands such as FREE PACKAGE and BIND PLAN.

ADB5184E The run is ending due to an error with a non-SQL command. Additional messages may have been sent to the DD SYSTSPRT.

Explanation:
The program terminated because of an error with a command other than an SQL statement, such as a Db2 utility statement or a Db2 command.

System action:
Processing stops.

User response:
Check the job log for return codes for commands other than SQL statements and examine the error messages. Then, resolve the errors.

ADB5254I The SSID parameter that is passed to the program can not be validated. The information that is used for SSID validation can not be obtained because an IFI return code <rc> and a reason code <rc> occurred during the execution of the -DIS GROUP DETAIL DB2 command.

Explanation:
The -DIS GROUP DETAIL command fails, therefore no information can be used to validate the SSID parameter.

System action:
Processing continues.

User response:
Optionally, if the job fails, verify that the value in the SSID parameter is correct.

ADB5255I The SSID parameter that is passed to the program can not be

validated. The information that is used for SSID validation is not complete in the output of -DIS GROUP DETAIL. More information, than can be displayed, exists.

Explanation:
The maximum number of subgroup attachment groups is displayed in the output from executing the -DIS GROUP DETAIL DB2 command. More information exists but cannot be displayed. The SSID is passed to the program but is not validated.

System action:
Processing continues.

User response:
Optionally, if the job fails, verify that the value in the SSID parameter is correct.

ADB5256I **The SSID parameter that is passed to the program is not recognized as a DB2 subsystem name in a non-data sharing environment, or as a DB2 subsystem name for a member, group, or subgroup in a data sharing environment.**

Explanation:
The SSID parameter that is passed to the program does not match one of DB2 subsystem names, group attachment name or subgroup attachment names in the output from executing the -DIS GROUP DETAIL DB2 command. The SSID problem might cause the job to fail.

System action:
Processing continues.

User response:
Optionally, if the job fails, verify that the value in the SSID parameter is correct.

ADB5257I **Table(s) have been successfully recreated on the accelerator.**

Explanation:
The specified schema and table name have been successfully loaded.

System action:
Processing continues.

User response:
None.

ADB5258E **Table(s) cannot be recreated on the accelerator. An error occurred during the call to the accelerator stored procedure *procedure_name*.**

Explanation:
Unexpected error in accelerated table recreate processing.

System action:
Processing stops.

User response:
Investigate the failure reason and take remedial action.

ADB5262W **The REPLACE keyword in the LOAD control statement for table *tbname* is converted to RESUME YES. Reason *reason_code***

Explanation

LOAD REPLACE is not allowed for the reason listed in *reason_code*. Therefore, the REPLACE option was changed to the RESUME YES option in the LOAD utility statement.

tbname
Table name

reason_code
One of the following reasons:

01
The table to be loaded is a system-period temporal table with data versioning.

02
The table to be loaded is an archive-enabled table.

03
The table is in a multi-table table space.

System action:
Processing continues.

User response

No action is required.

ADB5263E **The program *program_name* invocation was not from an authorized program facility (APF) library.**

Explanation:
Only load modules from an APF task can invoke the APF protected supervisor calls (SVCs).

System action:
Processing stops.

User response

Ensure that the program is APF authorized by making changes to meet the following conditions:

- The steplib data set name matches the data set name in the APF list.
- Each data set in the concatenation is APF authorized.
- The APF list specifies the correct volid.

- When SMS is specified as the volid in the APF list, the volume shown in the LISTC output is SMS managed.
- The required module names are listed in the output from PARMLIB.

ADB5264W **Reason: *reason***

Explanation

Shows details for the preceding message.

reason
The reason for the error in the preceding message.

System action:
Processing stops.

User response:
Resolve the problem and re-run the job.

ADB5266E **Table(s) cannot be loaded in the accelerator.**

Explanation:
Unexpected error in the accelerated table load processing.

System action:
Processing stops.

User response:
Investigate the failure reason and take remedial action.

ADB5267E **Table(s) cannot be deleted from the accelerator.**

Explanation:
Unexpected error in the accelerated table delete processing.

System action:
Processing stops.

User response:
Investigate the failure reason and take remedial action.

ADB5269E **Table(s) cannot be added to the accelerator.**

Explanation:
Unexpected error in the accelerated table add processing.

System action:
Processing stops.

User response:
Investigate the failure reason and take remedial action.

ADB5271E **The accelerated tables cannot be table-set in the accelerator.**

Explanation:
Unexpected error in the accelerated table enable/disable processing.

System action:
Processing stops.

User response:
Investigate the failure reason and take remedial action.

ADB5274E **The table *table-name* cannot be archived in the accelerator.**

Explanation:
Unexpected error in the accelerated table archive processing.

System action:
Processing stops.

User response:
Investigate the failure reason and take remedial action.

ADB5277I **The following list of *n* table(s) were *operation* on the accelerator *accelerator-name*: *table-names***

Explanation

The requested operation was successful.

n
The number of tables on which the operation was performed.

operation
The operation that was performed, such as "loaded" or "enabled for replication."

accelerator-name
The name of the accelerator server.

table-names
The qualified names of the tables on which the operation was performed. Each table name is separated by a comma.

System action:
Processing continues.

User response

No action is required.

ADB5278E **User restart label *label_name* was not found. Review WSL and ensure to have the specified restart label.**

Explanation

ADBTEP2 could not restart the WSL, because the specified restart point does not exist in the WSL.

label_name
The string that was specified as the requested restart point.

System action:
Processing stops.

User response:
Correct the WSL so that it includes the label for the requested restart point, and then restart the WSL.

Related tasks

"Restarting a WSL" on page 556
If a work statement list (WSL) fails in the middle of execution, you can restart it at the point that it failed or at a point that you specify. A WSL uses the restart capability of ADBTEP2.

ADB5280E Information about table(s) on the accelerator cannot be retrieved.

Explanation:
An unexpected error occurred when trying to retrieve the accelerated table information.

System action:
Processing stops.

Programmer response:
Investigate the failure reason and take remedial action.

Related reference

"ADMIN ACCELERATOR GET TABLE INFO statement" on page 968
The ADMIN ACCELERATOR GET TABLE INFO statement retrieves accelerated table information that is used to restore its acceleration and replication status.

ADB5281I Information about table(s) on the accelerator has been retrieved.

Explanation:
The accelerated table information was successfully retrieved.

System action:
Processing continues.

Programmer response

No action is required.

Related reference

"ADMIN ACCELERATOR GET TABLE INFO statement" on page 968
The ADMIN ACCELERATOR GET TABLE INFO statement retrieves accelerated table information that is used to restore its acceleration and replication status.

ADB5250W A call to SYSPROC.ADMIN_INFO_SSID returned SQL code *sqlcode*. ADBTEP2 will use command -DIS GROUP DETAIL instead.

Explanation

Prior to using the SSID parameter that is passed to the program, it is compared to the SSID name returned by a call to stored procedure SYSPROC.ADMIN_INFO_SSID. However, this call failed, as described by the SQL code in this message.

sqlcode
The SQL return code (SQLCODE).

System action:
Processing continues with the program using the Db2 command `-DIS GROUP DETAIL` to find the SSID name instead.

User response:
Ensure stored procedure SYSPROC.ADMIN_INFO_SSID is properly installed.

Related information

SQL codes (Db2 12 for z/OS documentation)

-DISPLAY GROUP (Db2) (Db2 12 for z/OS documentation)

ADMIN_INFO_SSID stored procedure (Db2 12 for z/OS documentation)

ADB5299E An error occurred while processing the ADMIN UNLOAD statement for the image copy process.

Explanation

The image copy cannot be processed because the ADMIN UNLOAD failed. The possible cause of failure is indicated by the reason code. See the following list for an explanation of the reason code:

9995
The image copy database or table space was not found.

9996
The image copy destination was not found.

9997
The image copy date or time is in the wrong format.

9999
The ADMIN UNLOAD statement is incomplete or contains a syntax error.

System action:
Processing stops.

User response:

Correct the ADMIN UNLOAD statement according to the reason code and rerun the job.

ADB5300E **The pending changes action parameter (PCACT) is set to CANCEL. This prevents the change from being recovered when there are pending changes that modify the same or related objects.**
To recover this change and set the pending changes to DEFINED status, set the PCACT parameter to SUPERSEDE and re-submit the job (e.g. PCACT(SUPERSEDE)).

Explanation:
The change cannot be recovered when there are pending changes that modify the same or related objects. The pending changes action parameter (PCACT) is set to CANCEL.

System action:
Processing stops.

User response:
Complete the pending changes first, or change the PCACT parameter to SUPERSEDE. Then, resubmit the job.

ADB5411I **The RESTART AT statement is ignored and restart will occur at the restart point that is specified in the RESTART() parameter because the restart point name <string_name> in this statement was used previously.**

Explanation:
If the restart point name in RESTART AT statement was used previously and a different restart point name is specified in the RESTART() parameter, the RESTART() restart point overrides the RESTART AT restart point.

System action:
Processing continues.

User response:
None.

ADB5412E **The restart point name <string_name> in the RESTART AT statement was used previously.**

Explanation:
If the restart point name in the RESTART AT statement was used previously, the program will stop to prevent an unwanted second restart.

System action:
Processing stops

User response:
Specify parameter RESTART() with the used restart point name to confirm reusing the same restart point, or enter a blank value in the restart_label column of the checkpoint record.

ADB5413E **The restart point name <string_name> in the RESTART AT statement is different from the restart point name <string> in the RESTART() parameter.**

Explanation:
The program will stop if the restart point names in the RESTART AT statement and the RESTART() parameter do not match.

System action:
Processing stops.

User response:
Remove either the RESTART AT statement or remove the RESTART() parameter.

ADB5414W **Down-level APPLICATION COMPATIBILITY, app_level, is detected. Environmental variables that depend on the application level will not be retained for restart.**

Explanation:
The restart point cannot preserve application level sensitive environmental variables.

System action:
Processing stops.

User response:
Rebind ADBTEP2 and ADBTEPR package with correct APPLCOMPAT bind option if needed.

ADB5501E **The DDL file validation date has expired. Create timestamp: *timestamp*. Validation date: *date*.**

Explanation:
The statements that you can run with the auth-switch ID depend on your authority as defined in the RACF profile that protects the resource. If you have READ authority, the DDL must be run within 8 days of being created.

System action:
Processing stops.

User response:
Regenerate the DDL file and try again.

ADB5507E **Use of WSL auth-switching was rejected. The submitter does not have ALTER authority to use the RACF profile of <ID>.**

Explanation:

Use of WSL auth-switching requires the submitter to have ALTER authority to use the RACF profile.

System action:
Processing stops.

User response:
Verify the RACF facility setting of ADBAUTHS and ensure that the submitter has ALTER authority to use the auth-swith ID's profile.

ADB5700E **Unrecognized parameter in the ADBTEPIN DD card: *parameter_name***

Explanation:
An invalid parameter name was specified in the ADBTEPIN DD card.

System action:
Processing stops.

User response:
Remove or correct the invalid parameter name, and resubmit the job.

ADB5701E **Invalid value for parameter BINDERROR in the ADBTEPIN DD card: *parameter_value*. Valid values are MAXE, SAVE and IGNORE. If MAXE is specified, the value *n* that is specified for MAXE(n) parameter in the PARMS field of the DB2 RUN statement determines the number of DSN commands that can fail before the batch job is terminated.**

Explanation:
An invalid value was specified for parameter BINDERROR in the ADBTEPIN DD card.

System action:
Processing stops.

User response:
Correct the invalid value and resubmit the job.

ADB6001W **There is invalid text in file ALTPARM.**

System action:
None.

User response:
Correct the input parameter in ALTPARM and try again.

ADB6002E **The DD statement *DDstatement* is missing or is incorrect.**

System action:
Processing stops.

User response:
Supply the missing DD statement, and try again.

ADB6003E **Program ADBALT detected an ONCODE condition.**

System action:
A return code of 12 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB6025E **Program ADBALT detected an ONCODE condition.**

System action:
A return code of 12 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB6026E **Open input error: *text1*.**

System action:
Processing stops.

User response:
Correct the open input error for CHGIN and resubmit the job.

ADB6027E **Close input error: *text1*.**

System action:
Processing stops.

User response:
Correct the close input error for CHGIN and resubmit the job.

ADB6041W **There is invalid text in file *CPPARM*.**

System action:
None.

User response:
Correct the input parameter in CPPARM and try again.

ADB6042E **For one-to-many copy privileges, the specified version scope *version scope qualifier version scope name* definition might be empty or the *NAMES* does not have any requests to generate GRANT DDLs for the copy privileges command.**

System action:
None.

User response:
The objects lists generated for processing copy privileges might be empty or the specified version scope definition for one-to-many might be empty or incorrect.

ADB6043E **The source object type specified to copy privileges is invalid.**

System action:
Processing stops.

User response:
Correct the source type and the try again.

ADB6044E **There are empty input parameters in file *CPPARM*.**

System action:
Processing stops.

User response:
Specify input parameters in file *CPPARM* to complete the copy privilege run.

ADB6045E **The catalog row stack is full and the run will terminate.**

System action:
Processing stops.

User response:
The copy privileges command for one-to-many can accommodate a maximum of 12500 GRANTS for source objects. Contact IBM Software Support.

ADB6046W **For one-to-many copy privileges, the specified quick scope or version scope *<version scope qualifier>. <version scope name>* does not have objects that match the specified FROM type *<FROM object type>*. An empty definition will result in no generated GRANT DDLs for the copy privileges command.**

System action:
Processing continues.

User response:
The specified TO version scope or quick scope could not find the objects that match the specified FROM type. This results in no GRANTs generated and can lead to an empty file.

ADB6300E **Processing error. The program will now terminate.**

Explanation:
An error occurred in processing.

System action:
Processing stops.

ADB6310I **No LOAD utility options specified.**

Explanation:
LOAD utility options missing.

System action:
Processing stops.

User response:
Supply the LOAD utility option, and try again.

ADB6311E **The null indicator is set to *value* in the HPU configuration, which does not match the default value. Only the default setting is allowed when data conversion is involved.**

Explanation:
HPU is used as the unload method, and the HPU PARMLIB parameter VUU014/ULNULL is set to a value that does not match the default value, FF00.

System action:
Processing stops.

User response:
Change the configuration of HPU to use the default null indicator and rerun the job.

ADB7001W **The REPLACE keyword in the LOAD control statement for table *table_name* is converted to RESUME YES. Reason: *reason_code*.**

Explanation

DB2 restrictions on LOAD REPLACE require a change to the LOAD control statement. The reason code indicates the source of the error:

01
The table to be loaded is a system-period temporal table with data versioning define.

02
The table to be loaded is an archive-enabled table.

03
The table is under a multi-table table space and not all the tables under the table space are migrated.

System action:
Processing continues.

User response:
Review the LOAD control statement for the specified table, particularly the REPLACE keyword. Correct the statement, if necessary, and try again.

ADB7002W **The LOAD REPLACE option is applied to the multi-table table space *ts_name*. Any additional tables in the target table space are left empty after migration.**

Explanation:
The LOAD REPLACE option is applied to the table space as specified in the LOAD Utility options because all the tables under the table space are selected for migration on the source system. Any additional tables in the table space on the target system are left empty after migration because the LOAD REPLACE option is used.

System action:

Processing continues.

User response:
Confirm that it is appropriate to use the LOAD REPLACE option before submitting the target jobs.

ADB7100E SQL statement too long - internal error

System action:
Processing stops.

User response:
Fix the problem and try again.

ADB7102E The table *table_name* contains too many columns.

Explanation:
You can assign up to 750 columns for a non-dependent table. Dependent tables can have up to 749 columns.

System action:
Processing stops.

User response:
Limit the number of columns to allowed values and try the operation again.

ADB7103E If ignore partitioning is specified, Object Compare will take partition information from the target. Partitioning is not allowed on partition-by-growth tablespace.

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7104W The *file_name* data definition is not defined, which can lead to errors due to insufficient sort work file size.

Explanation:
The *file_name* data definition (DD) file, which is an alternate location to store the record count, is not defined. Sometimes use of the *<version file name>* DD file can cause the sort process to underestimate the number of records in the file. Errors can occur due to insufficient sort work file sizes. Take action if the *file_name* is a version file created by GEN or DTC. If the version file is created from change management, you can ignore this information.

System action:
Processing stops.

User response:
The *file_name* DD is not defined, generate the job again. If the problem persists, make sure that the skeletons are current.

ADB7105E Substring outside string - internal error.

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7106W *<insert1>* source authorizations for *<insert2>* *<insert3>* will not be copied to the target because the grantor and grantee are the same. The problem is likely caused by masking.

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7107I *<insert1>* is an index on auxiliary table. It will be kept because the base table *<insert2>* is kept.

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7108E The explicit LOB table space *<insert1>* is still associated with auxiliary table *<insert2>* and therefore cannot be dropped.

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7109I The explicit LOB table space *<insert1>* is supposed to be dropped but is kept because the base table *<insert2>* is kept.

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7110I Change to clustering attribute is ignored because <insert1> can not be clustered.

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7112I *<insert1>* is table partitioned, *<insert2>* is *<insert3>*.

Explanation:
The table is being changed either from partitioned to non-partitioned table or vice versa.

System action:
Processing continues.

User response:
No action is required.

ADB7113I ***<insert1>* change is ignored because the ignore option *<insert2>* is specified.**

Explanation:
The change is part of the ignore fields specification which is part of this compare run.

System action:
Processing continues.

ADB7114W **Base table space has been changed from partitioned to PBG .<insert1> table space will be recreated with DB2 default values.**

Explanation:
Changing type of the table space to PBG causes the table space and table to be dropped and recreated. All implicit LOB and XML table spaces will be dropped and recreated by DB2 with attributes having default values.

System action:
Processing continues.

User response:
No action required.

ADB7115E **Invalid parent key of table.**

Explanation:
The referenced parent key has not been defined as a primary key or a unique key.

System action:
Processing stops.

User response:
Ensure that the parent key is defined as a primary key or a unique key.

ADB7116E **No match to the child column was found in the corresponding parent table.**

Explanation:
The referenced parent key does not have the same number of columns as the child key.

System action:
Processing stops.

User response:
Ensure that each child column as a corresponding parent column

ADB7117W **No index was created for the foreign key column.**

Explanation:
If the foreign key column is not indexed, the performance of DELETE on the parent table may be affected.

System action:
Processing continues.

User response:
For optimum performance, create an index for the foreign key column.

ADB7118W **The *<insert1>* table *<insert2>* is not in the current scope of analysis. The correctness of foreign key cannot be determined.**

System action:
Processing continues.

User response:
Verify that the parent table is in the catalog.

ADB7119E **Index extension type not supported. Type = *extension_type* Index= *index_creator.index_name***

Explanation:
The specified index that you are attempting to compare is a spatial index. Object Compare does not support spatial indexes. In this case, the table on which the index is defined has been dropped and recreated. This recreation might compromise existing column information for the spatial index.

System action:
Processing stops.

User response:
Specify a valid index for the comparison.

ADB7120I **The change has been ignored.**

Explanation:
The change is part of the ignore changes specification which is part of this compare run.

System action:
Processing continues.

User response:
No action is required.

ADB7121I **The ARRAYINDEXTYPEID of the source and target are different. *var_name* cannot be ignored.**

Explanation:
The index type of an associative array must be VARCHAR or INTEGER. If the source and target have different index types, then the index length and index subtype cannot be ignored.

System action:
Processing continues.

User response:
No action is required.

ADB7122I The change has been ignored because it is related to ignored *<insert1>* change.

Explanation:
The change is related to other changes which are part of the ignore specification included in this compare run.

System action:
Processing continues.

User response:
No action is required.

ADB7122I The change has been ignored because it is related to ignored *<insert1>* change.

Explanation:
The change is related to other changes which are part of the ignore specification included in this compare run.

System action:
Processing continues.

User response:
No action is required.

ADB7123E The logging attribute of the LOB table space *lob_tsname* that is associated with the base table *tbname* can not be changed to LOGGED because the logging attribute of the base table space *base_tsname* is NOT LOGGED.

Explanation:
If the logging attribute of the base table space is NOT LOGGED, the logging attribute of the LOB table space associated with the base table can not be LOGGED.

System action:
Processing continues.

User response:
Change the logging attribute of the base table space or the LOB table space and try again.

ADB7124I The logging attribute of the LOB table space *lob_tsname* that is associated with the base table *tbname* is changed to LOGGED. Information on the logging attribute of the base table space is not available.

Explanation:
An inconsistency exists if the logging attribute of the base table space is NOT LOGGED and the logging attribute of the LOB table space that is associated with the base table is LOGGED.

System action:
Processing continues.

User response:
If needed, change the logging attribute of the base table space or the LOB table space and try again.

ADB7125W Active versioning is in effect with this base table. In order to add a clone table the base table must be dropped and recreated.

Explanation:
The ALTER statement cannot be used to define a clone table because the base table is in a state of active versioning. Dropping and recreating the table will reset the active versioning and therefore allow you to add the clone table.

System action:
Processing continues.

User response:
None.

ADB7126I The active versioning that is in effect with this base table is caused by altering the table during this compare run. In order to add a clone table the base table must be dropped and recreated.

Explanation:
The ALTER statement cannot be used to define a clone table because the base table is in a state of active versioning. Dropping and recreating the table will reset the active versioning and therefore allow you to add the clone table.

System action:
Processing continues.

User response:
If needed, change the logging attribute of the base table space or the LOB table space and try again.

ADB7127W The information about base table active versioning is not available. Check whether the base table is in a state of active versioning before applying changes.

Explanation:
The ALTER statement cannot be used to define a clone table if the base table is in a state of active versioning. Because the target is a data set, the information about the active versioning is not available. The clone table will be added via an ALTER TABLE statement without dropping the base table.

System action:
Processing continues.

User response:
Check whether the base table has an active versioning before applying changes. If active versioning is in effect, reset the versioning by executing REORG followed by MODIFY RECOVERY. Then, run compare. Alternatively, you can run compare using the catalog as the target.

ADB7129W **Index extension type not supported. Type = *extension_type* Index= *index_creator*.*index_name***

Explanation:
The specified index that you are attempting to compare is a spatial index. Object Compare does not support spatial indexes. In this case, the table on which the index is defined has been altered or has no changes.

System action:
The specified index is skipped, and processing continues.

User response:
None.

ADB7130W **Clone table *<insert1>. <insert2>* is specified in the exclude specification. It will be *<insert3>*.**

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7131W **Clone table *<insert1>. <insert2>* is specified in exclude specification. It will not be *<insert3>.***

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7132W ***<Insert1> <insert2>. <insert3>* is specified in the *<insert4>* exclude specification. This object is excluded.**

System action:
Processing continues.

ADB7133W ***<Insert1> <insert2>. <insert3>* is excluded.**

System action:
Processing continues.

ADB7134W **History table *<insert1>. <insert2>* is specified in exclude specification.**

System action:
Processing continues.

ADB7135W **Temporal table *<insert1>. <insert2>* and history table are both excluded.**

System action:
Processing continues.

ADB7136E ***<insert1>. <insert2>* is an excluded object and needs to be implicitly dropped. To allow implicit drop of an excluded object, specify NO. Object Compare is terminated.**

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7137I **Exclude specification *<insert1>. <insert2>* can not be found.**

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7138I ***<insert1> <insert2>. <insert3>* is an excluded object. It will not be dropped.**

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7139E **The CREATE INDEX statement may lead to error SQLCODEN662/ SQLCODE -662 because the table *table_name* on which the index is being created is in the table space *tablespace_name*. The table space is defined as *tablespace_type*.**

Explanation:
The partitioned index cannot be created on the specified table space, or the table space cannot be index-controlled.

System action:
Processing continues.

User response

Verify that you are using the correct table space type for creating a partitioned index. Any changes to the table space type may be due to one of the following conditions:

1. Original definition of the table space was incorrect.

2. Changes to table space attributes SEGSIZE, MAXPARTITIONS, or NUMPARTS were specified.
3. Mask or ignore was specified on table space attributes SEGSIZE, MAXPARTITIONS, or NUMPARTS.
4. Generic ignore PARTITIONING field was specified.

ADB7140E ***<insert1> <insert2>* is specified in *<insert3>* exclude specification. This object is excluded.**

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7141I ***<insert1> <insert2>* is an excluded object. It will not be dropped.**

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7142I **Implicit XML Tablespace for target table *<insert1>. <insert2>* is excluded because its target table is excluded.**

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7143I **Implicit index *<insert1>* for target table *<insert2>. <insert3>* is excluded because its target table is excluded..**

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7144W **The *<insert1>* attribute of the implicit table space *<insert2>* cannot be altered to retain the specification from *<insert3>* because of *<insert4>*.**

Explanation:
When either of the tables being compared uses implicit table space, Object Comparison Tool or Administration Tool compares the DB2 default values against the original values from the source or target in order to preserve the attributes of the implicit table space. ALTER statements are generated for the differences. This message is displayed when the attribute referenced in the message cannot be altered on the DB2 release that Object Comparison Tool or Administration Tool is running on.

System action:
Processing continues.

User response:
The attribute cannot be altered. To alter the attribute, you must establish a DDL with explicit objects or migrate to a DB2 version that supports the alter.

ADB7145I **Column *column_name* cannot be altered by the ALTER TABLE ALTER COLUMN SET WITH DEFAULT statement. Reason code *reason_code*.**

Explanation

The table cannot be altered due to DB2 restrictions. The table will be dropped and re-created. See the following list for an explanation of the reason code:

1
The table must not be referenced by a view or a Materialized Query Table (MQT).

2
For LOB columns, only the default for inline LOB columns can be changed. The new default length cannot be greater than the inline length.

System action:
Processing continues.

User response:
No action is required.

ADB7146I **Table *table_name* will be recreated in IBM DB2 Analytics Accelerator.**

Explanation:
This message is issued when a table has been modified and will be offloaded to the IBM DB2 Analytics Accelerator to improve performance.

System action:
Processing continues.

User response:
No action is required.

ADB7147I **The tablespace *table_space* is a LOB tablespace, which can be dropped only after the auxiliary table has been dropped.**

Explanation:
A comparison of the LOB table spaces shows that the LOB table space needs to be dropped. Object Comparison tool generates the drop statement only when the auxiliary table is dropped because a LOB table space cannot be dropped when an association exists between it and an auxiliary table.

System action:
Processing continues.

User response:
If necessary, fix the problem and try again.

ADB7148I **Tablespace *table_space* is a LOB tablespace. Because the KEEPTGT option was specified, the tablespace will be kept even if it is not associated with an auxiliary table.**

Explanation:
When the KEEPTGT option is specified, Object Comparison tool will keep the LOB table space which exists on the target but not on the source. Even if the LOB table space is not associated with any auxiliary table after the changes are applied, the LOB table space is still kept.

System action:
Processing continues.

User response:
If necessary, fix the problem and try again.

ADB7149E **The table: *table.table* is partitioned and cannot be dropped explicitly. You can drop the table by dropping the table space *table.space*.**

Explanation:
Because the table space is excluded from the compare process, the table space cannot be dropped. Object Compare is terminated.

System action:
Processing stops.

User response:
Remove the exclusion on the table space and try the operation again.

ADB7150E ***statement table_name* to *subsystem* with privileges *privileges* is specified in *target_name* exclude specification. This grant is excluded.**

Explanation:
?

System action:
Processing continues.

User response:
?

ADB7151W ***statement table_name* to *subsystem* with privileges *privileges* is specified in *target_name* exclude specification. Excluding target grants has no effect because Object Comparison tool always attempts to retain the target grants.**

Explanation:
?

System action:
Processing continues.

User response:
?

ADB7153W ***schema.objectname* has been masked as *new-schema.new-objectname* during this compare run.**

Explanation

The qualified object name is masked for this comparison.

schema.objectname
The original qualified object name.

new-schema.new-objectname
The qualified object name with the mask applied.

System action:
Processing continues.

User response

No action is required.

ADB7154W **The dependent object information is needed. The version files must have a release marker of at least 814. Generate new version files with the current product JCL.**

Explanation:
This message is issued if the Object Compare processing requires object dependency information and at least one function, stored procedure , or trigger. It is used to determine the order these objects must be dropped and there is no change to dependency. If either the source or target is a version file with object dependency information, then the object dependency information from that version file will be used.

System action:
No system action is taken.

User response:
If a version file is not at or above the 814 level, it is recommended that the version file be regenerated using product JCL at the current level.

ADB7155E **The name *<consname>* of the *<constype>* constraint on the *<tbfname>* table is a duplicate name of another *<dupctype>***

constraint which was previously specified on the same table.

Explanation:
The constraint name must be different from the names of any referential, check, primary key, or unique key constraints previously specified on the table.

System action:
Processing stops.

User response:
Fix the problem by removing the duplicate name from the constraint definition or renaming the constraint with an unique name, and then try again.

ADB7156E **The *ts-type1* tablespace has more than one table. Changing the tablespace to a *ts-type2* tablespace will fail. Number of tables is: *num_tables***

Explanation

The table space cannot be converted to the indicated type, because it contains more than one table.

ts-type1
The current table space type.

ts-type2
The table space type to which Db2 Admin Tool is trying to convert the table space.

num-tables
The number of tables in the current table space.

System action:
Processing stops.

User response

If the failure occurred because a multi-table segmented table space in the compare target could not be converted to a partition-by-growth (PBG) table space, use a different name for the PBG table space in the compare source.

For example, if the compare target and source have the following objects, this error occurs:

Target	Source
Segmented table space A, which contains tables TB1 TB2 TB3	PBG table space **A**, which contains table TB1 PBG table space B, which contains table TB2 PBG table space C, which contains table TB3

To avoid this error, the source PBG table space must have a name other than A. For example, if the source and target contains the following objects, Object Compare can generate the change file to convert the target segmented table space to a PBG table space:

Target	Source
Segmented table space A, which contains tables TB1 TB2 TB3	PBG table space **D**, which contains table TB1 PBG table space B, which contains table TB2 PBG table space C, which contains table TB3

ADB7158E **The user-defined function *<function_name>* from *<origin>* is a *<function_type>*.**

Explanation:
This message is displayed when the compared objects include the non-inline SQL scalar function or the SQL table function, and the compare option **Bypass SQL PL functions** (parameter **BYPASSSQLPL**) is not set to Yes.

System action:
Processing stops.

User response:
Remove the reported SQL PL function from the source or the target and try again. Alternatively, you can set the compare option **Bypass SQL PL functions**to YES and try again.

ADB7157W **The *<insert1>* tablespace has more than one table. Changing the tablespace to a *<insert2>*tablespace may fail. Number of tables is: *<insert3>***

System action:
Processing continues.

User response:
Fix the problem and try again.

ADB7158I **One or more attributes of the implicit *<obj_type>* for table *<tbname>* *<colname>* *<part>* are**

altered to retain the *<srctgt>* value.

Explanation:
When a table is dropped and re-created or is added to the target system, DB2 creates implicit table spaces and indexes for the table with attributes that have default values. ALTER statements are generated to change the default values to the original target values or to the values from the source objects.

System action:
Processing continues.

User response:
No action is required.

ADB7159I **Field *<field_name>* changed from *<tgt_value>* to *<src_value>* but no ALTER statement is generated because the new value is the same as the default value from target system.**

Explanation:
No ALTER statement is generated because the implicit object is dropped and re-created with the attributes that have a default value that matches the value on the source system.

System action:
Processing continues.

User response:
No action is required.

ADB7160W **The *table_name* table was not placed in read-only mode before the *trigger_name* dependent trigger was dropped because the database or table space name of the table is unknown. Data integrity issues can occur during the change of the trigger.**

Explanation

The database name or table space name is unknown for one or more of the following reasons:

- The target is from DDL, and the table on which the trigger is built is not included in the compared objects.
- The target is from DDL, and the table on which the trigger is built is included in the compared objects. However, the table is created in an implicit database and table space.
- A previously created version file, which does not contain database and table space information for the trigger, is being used as the target.

System action:
Processing continues.

User response

Review the message. If necessary, take the appropriate action and resubmit the compare job.

If the target is from a DDL file, ensure that the table definition is included in the DDL file, and that the table is not created in an implicit database and table space. Otherwise, extract the target definition from the DB2 catalog by specifying the target from the DB2 catalog on the Compare panel. If a previously created target version file is being used, regenerate the version file.

ADB7161W **The tablespace uses index-controlled partitioning and has changed from non-large to large. Data in the last partition of the table might be discarded if the partitioning limit key is not set to the highest possible value for an ascending index key column or set to the lowest possible value for a descending index key column.**

System action:
Processing continues.

User response:
Review the generated APPLY jobs or WSL before applying the change. If necessary, update the limit key of the last partition to avoid the possibility of discarded data during the LOAD phase.

ADB7162W **The number of auxiliary tables associated with the source table might not be consistent with the number of LOB columns in the source table. Implicit LOB objects are used when auxiliary tables are re-created. After changes are applied, ensure that one auxiliary table exists for each LOB column.**

Explanation:
For tables that contain LOB columns, DB2 requires that LOB table spaces and auxiliary tables be created to hold the LOB data. When the base table is non-partitioned, DB2 requires one LOB table space and one auxiliary table be created for each LOB column. Object Comparison Tool checks whether the LOB objects definitions on the source agree with DB2 rules. This message is displayed when an inconsistency is found. Object Comparison Tool will re-create the LOB objects implicitly if the table is re-created. When the source comes from DDL file and implicit LOB objects are used, the version file generated from the DDL file might not contain enough information for Object Compare to determine the correctness of the LOB objects definitions.

System action:
Processing continues.

User response:
After processing completes, assess whether auxiliary table definitions are missing or if implicit LOB objects have been created. If table definitions are missing, fix the problem and try again.

ADB7164W **The logging attribute for *<obj_desc> <objname_v>* is unknown because the table space is not included in the compared objects. Ensure that the correct SHRLEVEL option is used for the REORG utility.**

Explanation:
SHRLEVEL CHANGE or SHRLEVEL REFERENCE REORG might not be executable on a NOT LOGGED table space because of DB2 restrictions. After the change, when the table space is NOT LOGGED, Object Compare will convert the SHRLEVEL option to a valid value if the user-specified SHRLEVEL is not applicable. This message is displayed when the logging attribute of a table space is unknown because the table space is not included in the compared objects. Ensure the correct SHRLEVEL option is used in the APPLY jobs for the REORG utility.

System action:
Processing continues.

User response:
Review the message. If necessary, fix the problem and try again.

ADB7163W **The number of auxiliary tables associated with the source table might not be consistent with the number of LOB columns in the source table multiplied by the number of partitions in the table space. Implicit LOB objects are used when the base table is re-created. After changes are applied, ensure that one auxiliary table exists for each LOB column in each partition.**

Explanation:
For tables that contain LOB columns, DB2 requires that table spaces and auxiliary tables be created to hold the LOB data. When the base table is partitioned, DB2 requires one LOB table space and one auxiliary table be created for each LOB column in each partition. Object Comparison Tool checks whether the LOB objects definitions on the source agree with DB2 rules. Because of apparent inconsistency, Object Comparison Tool re-creates the LOB objects implicitly if the table is re-created. When the source comes from DDL file and implicit LOB objects are used, the version file generated from the DDL file might not contain enough information for Object Compare to determine the correctness of the LOB objects definitions. Check and ensure the correctness of the source DDL file.

System action:
Processing continues.

User response:
After processing completes, assess whether auxiliary table definitions are missing or if implicit LOB objects have been created. If table definitions are missing, fix the problem and try again.

ADB7165I **ALTER is not allowed by DB2 for this operation because *<reason_v>*.**

Explanation:
ALTER is not allowed for this change because of DB2 restrictions. The object will be dropped and re-created.

System action:
Processing continues.

User response:
Review the message to determine the reason ALTER is not allowed.

ADB7166E **The EDITPROC is not valid for this table because of DB2 restrictions.**

Explanation:
The EDITPROC is not valid because of one of the following reasons: 1) The table contains LOB columns, 2) The table cannot have a ROWID, Identity, SECURITY LABEL or XML column when the WITH ROW ATTRIBUTES option is specified, 3) Column names cannot be more than 18 EBCDIC SBCD characters in length when the WITH ROW ATTRIBUTES option is specified.

System action:
Processing stops.

User response:
Correct the definitions of the table column.

ADB7167W **The *<attribute>* of the implicit XML table space changed from *<value_1>* to *<value_2>*. However, no change statement will be generated because the ALTER statement cannot apply the change.**

Explanation:
The attribute cannot be altered on the Db2 version that Object Comparison Tool is running on. The Db2 version is earlier than Db2 10 new-function mode.

System action:

Processing continues.

User response:
You cannot alter the attribute on this version. To alter the attribute, you must migrate to a Db2 version that supports the enhanced ALTER statement.

ADB7168E The source table space cannot contain the table record length. Specify a larger buffer pool to ensure that the page size is suitable for the table record length and that the table space can contain the record.

Explanation:
This change cannot be applied until you choose a proper buffer pool for the table space.

System action:
Processing stops.

User response:
Choose a proper buffer pool for the table space before applying the changes to the table.

ADB7169W The page size of the table space is unknown because the table space is not included in the compared objects. Ensure that the row length for the table does not exceed the page size limit.

Explanation:
Object Compare checks that the row length of the table does not exceed the page size limit. This message is displayed when Object Compare cannot determine the page size of the table space because the table space is not included in compared objects.

System action:
Processing continues.

User response:
Review the message. Ensure that the table space is specified in compared objects. If necessary, specify a buffer pool with proper page size before running the apply jobs.

ADB7170I Partitioning changes are ignored. LOB objects related to *<insert1>* partitions are not *<insert2>*.

Explanation:
The general ignore option PARTITIONING was specified for this compare run therefore all changes related to partitioning are ignored. Explicit LOB objects for added partitions are not created. Explicit LOB objects for dropped partitions are not dropped.

System action:
Processing continues.

User response:
No action is required.

ADB7171W The source contains an incomplete set of explicit LOB objects therefore all LOB objects for this base table will be created implicitly.

Explanation:
For tables that contain LOB columns, DB2 requires that LOB table spaces, auxiliary tables, and their indexes be created to contain the LOB data. When the base table is created all LOB objects must be created either explicitly or implicitly. Because the source contains explicit definitions for some of the LOB objects, and is missing the definition of other LOB objects, Object Compare creates implicitly all LOB objects for this base table.

System action:
Processing continues.

User response:
No action is required. However, if you want all LOB objects to be explicit, add the missing definitions and run compare again.

ADB7172W A single partition or multiple partitions were added by altering the table. New LOB objects for added partitions are created implicitly by DB2. Any explicit definitions of new LOB objects are ignored.

Explanation:
If partitions are added to a table using the ALTER TABLE ADD PARTITION statement and the table is in a PBG table space, then DB2 creates all needed LOB objects (LOB table space, auxiliary table, index on auxiliary table) for the partitions implicitly. Therefore explicit definitions for the LOB objects specified on the source can not be used.

System action:
Processing continues.

User response:
No action is required.

ADB7173I The SEGSIZE will be set to the default value 32 after a change of MAXPARTITIONS is applied.

Explanation:
While changes to MAXPARTITIONS are still pending, the SEGSIZE of a simple or segmented table space will not change. Once changes to MAXPARTITIONS are applied, the SEGSIZE will be set to the default value 32 by DB2. Therefore, Compare will not generate a statement for changing SEGSIZE to 32.

System action:

Processing continues.

User response:
No action is required

ADB7174W Archive table *<insert1>. <insert2>* is specified in *<insert3>* exclude specification.

System action:
Processing continues.

User response:
No action is required.

ADB7175W Archive-enabled table *<insert1>. <insert2>* and archive table are both excluded.

System action:
Processing continues.

User response:
No action is required.

ADB7176E Synonym *syn_name* for *syn_creator* is also found as a *obj_type*.

Explanation:
The name of the synonym has already been found as another *obj_type* in the target.

System action:
Processing stops.

User response:
Refer to the compare report to correct this error and rerun the job.

ADB7176I The table will be loaded to IBM DB2 Analytics Accelerator.

Explanation:
This message is issued when a table has been created and its data will be loaded to IBM DB2 Analytics Accelerator

System action:
Processing continues.

User response:
None.

ADB7177E *Obj_type obj_creator.obj_name* is also found as a *obj_type*.

Explanation:
Obj_type obj_creator.obj_name has already been found as another *obj_type* in the target.

System action:
Processing stops.

User response:
Refer to the compare report to correct this error and rerun the job.

ADB7177I The table will be removed from IBM DB2 Analytics Accelerator.

Explanation:
This message is issued when a table has been dropped and will be removed from the IBM DB2 Analytics Accelerator.

System action:
Processing continues.

User response:
None.

ADB7180E User Defined SQL Scalar Function *<insert1>* has versions with different SECURED options.

Explanation:
All versions of a SQL Scalar function must be all SECURED or all NOT SECURED.

System action:
Processing continues. Object Compare ends with RC=8.

User response:
Make all versions of the procedure consistent and try again.

ADB7181E Native Stored Procedure *procedure* has versions with different COMMIT ON RETURN options.

Explanation:
The versions of a Native Stored Procedure must be all COMMIT ON RETURN or AUTONOMOUS.

System action:
Processing continues. Object Compare ends with RC=8.

User response:
Make all versions of the procedure consistent and try again.

ADB7182E Source Procedure *<insert1>* type is *<insert2>* and target procedure *<insert3>* type is *<insert4>*. To compare native stored procedures, both source and target procedures must be the same type.

Explanation:
Native stored procedures are compared only when both source and target are of the same type.

System action:
Processing continues. Object Compare ends with RC=8.

User response:
Correct source and target procedures so that the procedure type is the same and then try again.

ADB7183E **OMPRESS is specified as YES, therefore the index is changed to use index compression. The buffer pool *<insert1>* must be 8 KB, 16 KB, or 32 KB in size.**

Explanation:
If compress is changed to YES, then the size of buffer pool must be 8K, 16K or 32K.

System action:
Processing continues.

User response:
Make sure the buffer pool size is correct.

ADB7184E **A duplicate record was detected for object *<object name>*. The error is probably caused by the renaming of an object or by masking from *<target object name>* to *<source object name>*. The run will terminate.**

Explanation:
A duplicate record error occurred. The error was probably caused by an attempt to rename an object or to mask.

System action:
Processing stops. A return code of 12 is generated for the Object Compare step.

User response:
Change the name value so that the rename or mask is no longer a duplicate. Then, try run the job again.

ADB7185W **User-defined SQL Scalar Function *<insert1>* has versions with the same name but different signatures. This may cause problems when changes are applied.**

System action:
Processing continues. Object Compare ends with RC=4.

User response:
Make all versions of the function consistent and try again.

ADB7186I **Column *<colname_v>* is referenced by triggers. The column can not be altered.**

System action:
Processing continues.

ADB7187E **The version level *version_level* in the *source_target* version file is not supported.**

Explanation:
An unsupported version was detected in the version file. The version file was created by a prior release and is not supported.

System action:
Return code 8 is set and processing is halted.

User response:
Examine the version in the version file. Create the version file again using the current release, or convert the version file to the current release.

ADB7188W **The *<source_target>* version file has an unknown version level.**

Explanation:
An unknown version was detected in the version file. The version file was created by a prior release.

System action:
Return code 4 is set and processing continues.

User response:
Verify that the unknown base version record is valid.

ADB7190I **Trigger will be dropped and re-created because of change to referenced column(s).**

System action:
Processing continues.

ADB77191I **Column *column_name* cannot be dropped by the ALTER TABLE DROP COLUMN RESTRICT statement. Reason code: *reason_code*.**

Explanation

The specified column cannot be dropped.

column_name
The name of the column.

reason_code
A reason code that indicates why the column cannot be dropped:

1
The column is the only column in the table.

2
The column has a security label defined.

3
The column is a DOCID column.

4
The column is a hidden ROWID column.

5
The column is a ROWID column, and a LOB column is dependent on it.

6
The column is defined as ROWID GENERATED BY DEFAULT, and the table contains a hidden ROWID column.

7
The column is part of the table partitioning key.

8
The column is part of the hash key.

9
The remaining columns in the table are all hidden.

10
The column is referenced in the definition of a period.

11
The column is an XML column.

12
The column is referenced by views, indexes, triggers, row permission, column mask, or inline SQL_table functions.

13
The column contains check constraints.

14
The column contains unique constraints.

15
The column contains referential constraints.

System action:
Processing continues.

User response:
Remove the column restriction or dependency and try again.

Related reference
ALTER TABLE (Db2 12 for z/OS)

ADB7192I **ALTER TABLE DROP COLUMN RESTRICT statement cannot be generated for table *table_name*. Reason code *reason_code*.**

Explanation

The column cannot be dropped. See the following list for an explanation of the reason code:

1
There are triggers defined on the table.

2
The table space is not a universal table space (UTS).

3
The table is a system-period temporal table.

4
The table contains check constraints.

5
The table is a created global temporary table.

6
The table is a history table.

7
The table has an edit procedure or validation exit procedure.

8
The table is referenced by extended indexes, materialized query tables, or inline SQL table functions dependency.

9
The table is referenced by row permissions or column masks.

10
There are INSTEAD OF triggers defined on a view that is dependent on the table.

11
If columns are dropped from tables that have clones, the clone tables cannot be added again.

System action:
Processing continues.

User response:
Remove the table, table space restriction, or dependency, and try again.

ADB7193I **(PC) *<insert1>* column *<colname>* dropped.**

Explanation:
There is an outstanding pending change to drop specific column. The column will be processed as if it was dropped.

System action:
Processing continues.

ADB7194S **Table *table_name* cannot include an identity column because it is a history table. Correct the definition for column *col_name*.**

Explanation:
A table is specified as a history table but the table definition is not valid because it contains an identity column.

System action:
Processing stops. The return code is 12.

User response:
Correct the column definition.

ADB7195E **Cannot Compare *source_table_type* *source_table_name* with *target_table_type* *target_table_name***

Explanation

The specified tables cannot be compared, because a required table is missing.

One of the following situations exist:

1. The source or the target contains a history table but not the associated temporal table. Both tables are required.
2. The source or the target contains an archive table but not the associated archive-enabled table. Both tables are required.

System action:
Processing stops.

User response:
Include the missing temporal or archive-enabled table.

ADB7196I **The column *column_name* cannot be added by the ALTER TABLE ADD COLUMN statement. Reason code: *reason_code*.**

Explanation

The specified column cannot be added.

column_name
The name of the column.

reason_code
A reason code that indicates why the column cannot be added:

1
The column is defined as NOT NULL but does not have a WITH DEFAULT clause specified.

System action:
Processing continues.

User response:
Correct the SQL statement and try again.

Related reference

ALTER TABLE (Db2 12 for z/OS)

ADB7198W **The Unload Altered Tables (unload_altered_tables/unldaltb) function was not activated for this compare process. Columns that are dropped from the table will not be unloaded and cannot be recreated.**

Explanation

Tables were not unloaded during the analyze process, because one of the following options was set to NO:

- the **Unload altered tables** option on the **Options for Change Functions (ADB2PCO)** panel
- the unload_altered_tables CM batch parameter

Because the data was not unloaded, any dropped columns cannot be recreated.

System action:
Processing continues.

User response:
None.

Related information

“UNLOAD_ALTERED_TABLES” on page 727

ADB7199E ***base_type base_name* cannot be dropped because *dep_type dep_name* depends on it.**

Explanation:
An object cannot be dropped when another object depends on it. The dependent object cannot be dropped because it is not included in the target. All objects that depend on another object must be in the target so that they can be dropped and re-created if required. Any attempt to drop an object with dependents is rejected by DB2.

System action:
Processing is halted and return code 12 is set for the step.

User response:
Include all dependent objects in the target. If you are using the ALT command to alter a table, use the ADD primary command from panel ADB27CA and add the objects that are referenced in this message. See *Altering or redefining a table with the ALT command* in the *DB2 Administration Tool for z/OS User's Guide and Reference*.

ADB7200I ***action_indicator old_option* changed to *new_option*.**

Explanation

The option has been changed. The *action_indicator* is one of the following:

- (A) - ALTER; the change will be implemented by the ALTER statement.
- (D) - DROP; the change will be implemented by dropping and recreating the object.

System action:
Processing continues.

User response:
No action is required.

ADB7201I ***action_indicator option* changed from *target_option* to *source_option***

Explanation

The option has been changed. The *action_indicator* is one of the following:

- (A) - ALTER; the change will be implemented by the ALTER statement.
- (D) - DROP; the change will be implemented by dropping and recreating the object.

System action:
Processing continues.

User response:
No action is required.

ADB7202I ***action_indicator* Default text *action*: *default_text***

Explanation

The option has been changed. The *action_indicator* is one of the following:

- (A) - ALTER; the change will be implemented by the ALTER statement.
- (D) - DROP; the change will be implemented by dropping and recreating the object.

Action is one of the following:

- added
- deleted

System action:
Processing continues.

User response:
No action is required.

ADB7203I **Grant(source): Grantor=*source_grantor_role source_grantor* Grantee:*source_grantee_role source_grantee* (Not propagated)**

Explanation:
If CMDELTA mode is not being used and if a compared object has new grants on the source that are not in the target, Object Comparison Tool will not propagate new grants from the source and will not generate any new source grant statements.

System action:
Processing continues.

User response:
No action is required.

ADB7204I **Grant(target): Grantor=*target_grantor_role target_grantor* Grantee:*target_grantee_role target_grantee* (Kept)**

Explanation:
If the grants exist on the target when the object is dropped and re-created, the target grants are kept.

System action:
Processing continues.

User response:
No action is required.

ADB7205E **The ALTER TABLE ADD VERSIONING statement cannot be processed, because the history table *history_table_qualifer.history_table_name* was not defined at the time the ADD VERSIONING statement was issued in the DDL file.**

Explanation:
The specified history table must exist before the ALTER TABLE ADD VERSIONING statement is issued.

System action:
Processing stops.

User response:
Correct the DDL. Make sure that the history table is defined before the ALTER TABLE ADD VERSIONING statement is issued.

ADB7206I **Grant(target): The subtype for character string columns (column type CHAR, VARCHAR, or CLOB) will be changed from SBCS to MIXED because the encoding scheme of the table is converted to UNICODE.**

Explanation:
Character data (CHAR, VARCHAR, and CLOB) is encoded in Unicode UTF-8, which DB2 considers to be mixed data by default.

System action:
Processing continues.

User response:
None.

ADB7206E **The column definition includes a CCSID attribute that can be specified only if the table has the EBCDIC encoding scheme.**

Explanation:
The column attribute CCSID 1208 or CCSID 1200 was specified for a column in a table with an encoding scheme that is not EBCDIC. In such cases, Object compare issues an error message to correct the problem to avoid run time failure.

System action:
Processing stops

User response:
Correct the encoding scheme for tables with EBCDIC to specify the CCSID attribute in the column definition. After the corrections are made, resubmit the job.

ADB7207E The column definition includes a CCSID attribute that is not allowed on a table that has EDITPROC or VALIDPROC defined on it.

Explanation:
The column attribute CCSID 1208 or CCSID 1200 was specified for a column in a table with EDITPROC or VALIDPROC defined on it. In such cases, Object Compare issues an error message to correct the problem to avoid run time failure.

System action:
Processing stops.

User response:
Correct the EDITPROC or VALIDPROC for tables to specify CCSID attribute in column definition. After the corrections are made, resubmit the job.

ADB7208E The column definition includes a CCSID clause and a FIELDPROC clause. Both clauses are mutually exclusive and are not allowed in the same column definition.

Explanation:
The column attribute CCSID 1208 or CCSID 1200 was specified for a column in a table with the FIELDPROC clause.

System action:
Processing stops.

User response:
Correct the FIELDPROC clause in the column definition to specify the CCSID attribute in the same column definition. After the corrections are made, resubmit the job.

ADB7205E The column definition includes a CCSID attribute that can be specified only if the table has the EBCDIC encoding scheme.

Explanation:
The column attribute CCSID 1208 or CCSID 1200 was specified for a column in a table with an encoding scheme that is not EBCDIC. In such cases, Object Compare issues an error message to correct the problem and avoid runtime failure.

System action:
Processing stops.

User response:
Correct the encoding scheme for tables to EBCDIC to specify the CCSID attribute in column definition. After the corrections are made, resubmit the job.

ADB7210I The START and RESTART WITH target values will not be changed because 'YES' was specified for the option 'Retain START and RESTART values for sequence object'.

Explanation:
If 'YES' is specified, the START value and RESTART WITH values of the target sequence will be retained and no ALTER SEQUENCE... RESTART statement will be generated. If 'NO' is specified and ignores for START and RESTART fields are not specified, the statement will be generated with values from source to make the target same as the source.

System action:
Processing continues.

User response:
No action is required.

ADB7211E Field *field-name* should be changed from *target-value* to *source-value* but option Allow PBR2 to PBR changes is set to "NO". The change will not be generated.

Explanation:
An attempt was made to change the value of the table space or partition attribute that is specified by *field-name* from *source-value* to *target-value*. However, on panel Options for Change Functions, field Allow PBR2 to PBR changes was set to NO, which prevents the PAGENUM table space attribute from being changed from RELATIVE to ABSOLUTE. The change to the *field-name* value cannot be made unless PAGENUM can be changed from RELATIVE to ABSOLUTE.

System action:
Processing stops.

User response:
To allow the change to the *field-name* value, change the value of Allow PBR2 to PBR changes to YES.

ADB7212E Before DB2 Version 12, option COMPRESS on implicit LOB table space is invalid.

Explanation:
Before Db2 12, option COMPRESS is invalid on an implicit LOB table space.

System action:
Processing stops.

User response:
Do not use option COMPRESS on an implicit LOB table space.

ADB7213E Option COMPRESS on LOB table space *table-space* is changed from NO to YES but the associated base table space is not a universal table space.

Explanation:
If the base table space is not a UTS, SQLCODE -763 is issued with the explanation that the table space is a LOB table space and COMPRESS YES was specified, but the associated base table space is not a universal table space.

System action:
Processing stops.

User response:
Change the type of the associated base table space to a universal table space.

ADB7215W The source value for option <option name> is omitted. The option will be changed to the default value.

Explanation:
A difference was found when comparing advanced trigger options. Since a new value was not specified a default value will be used instead.

System action:
Processing continues.

User response:
No action is required.

ADB7216I The source value for option *option_name* is omitted. Since target value *option_value* is the same as default the option will not be changed..

Explanation:
A difference was found when comparing advanced trigger options. Since a new value was not specified and existing value is the same as the default the option remains unchanged.

System action:
Processing continues.

User response:
No action is required.

ADB7221W The default value for appended column *<column_name>* could not be propagated from the temporal table to the history table.

Explanation:
The default value could not be propagated to the history table when adding a new column in the temporal table because DB2 for z/OS does not allow it.

System action:
Processing continues.

User response:
None

ADB7222W The difference in the default value for column <column_name> cannot be implemented in a history table.

Explanation:
The default value cannot be altered in the history table because DB2 for z/OS does not allow it.

System action:
Processing continues.

User response:
None

ADB7223I The table space *tablespace_name* is a history table space that can be dropped only after the history table has been dropped.

Explanation:
A comparison of the history table spaces shows that the history table space needs to be dropped. Object Comparison tool generates the drop statement only when the history table is dropped because a history table space cannot be dropped when the history table is associated with a temporal table.

System action:
Processing continues.

User response:
None.

ADB7224I The table space *tablespace_name* is a history table space. Because YES was specified for the Suppress DROP of objects option, the table space will be kept even if it is not associated with a history table.

Explanation:
When YES is specified for the Suppress DROP of objects option, Object Compare Tool will keep the history table space, which exists on the target but not on the source. Even if the history table space is not associated with a history table after the changes are applied, the history table space is still kept.

System action:
Processing continues.

User response:
None.

ADB7225I The table space *tablespace_name* is a history table space. Because YES was specified for the Suppress DROP of objects option, the table space will be kept even if it is not associated with a history table.

Explanation:
When YES is specified for the Suppress DROP of objects option, Object Compare Tool will keep the history table space, which exists on the target but not on the source. Even if the history table space is not associated with a history table after the changes are applied, the history table space is still kept.

System action:
Processing continues.

User response:
None.

ADB7226W **The referenced table or view *objschema.objname* is not included in this run or was dropped from the target.**

Explanation:
A table or view referenced by the object being processed is not included in this compare run. or was dropped from the target during the compare run.

System action:
Processing continues.

User response:
Check whether the missing table or view needs to be included in the compare run. If the missing object was dropped, update the trigger definition.

ADB7227E **A duplicate record for object *obj_schema.obj_name* was found. The object was not renamed, and its name was not masked. The run will terminate.**

Explanation:
An object with the same name as the name specified was found.

System action:
Processing stops.

User response:
Specify a different name for the object.

ADB7228W **INSERTALG column cannot be changed to 0. The target (TS:*database_name.tablespace_name*) INSERT ALGORITHM value remains *INSERTALG-value*.**

Explanation

The INSERT ALGORITHM attribute of the source table space is 0. However, Db2 for z/OS does not support altering INSERT ALGORITHM to 0. Therefore, this attribute of the target table space remains unchanged.

The INSERT ALGORITHM attribute value is stored in the INSERTALG column of the Db2 catalog table SYSTABLESPACE.

database_name.tablespace_name
The name of the target table space, qualified by the name of the database.

INSERTALG-value
The value of the INSERT ALGORITHM attribute of the target table space. Possible values are 1 or 2.

System action:
Processing continues.

User response

No action is required.

Related reference
SYSTABLESPACE catalog table (Db2 12 for z/OS)

ADB7229I **Cannot use ALTER statements to *operation* because *reason*. The *object-type* will be dropped and re-created.**

Explanation

When recovering a change, Db2 Admin Tool usually performs the specified operation by using ALTER statements. However, in this case, Db2 Admin Tool must drop and re-create the object.

operation
The operation. For example: drop a column from a table.

reason
The reason why the operation cannot be performed by using ALTER statements. For example: the table is part of an RI relationship.

object-type
The type of Db2 object. For example: table.

System action:
Processing continues.

User response

No action is required.

ADB7230W **Change to DSSIZE is not valid for this tablespace's partitions because of DB2 restrictions.**

Explanation:
Although the table spaces that are being compared have different partition DSSIZE values, Object Comparison Tool did not generate a change for DSSIZE, because changing the DSSIZE value would cause an error. The source table space uses absolute page numbering (the PAGENUM attribute is ABSOLUTE), and Db2 does not allow you to change the DSSIZE value if PAGENUM is ABSOLUTE.

System action:
Processing continues.

User response:
If you want the partition DSSIZE values of the target table space to be changed to match the partition DSSIZE values of the source table space, change the PAGENUM attribute of the source table space to RELATIVE. Otherwise, the DSSIZE values are not changed, and no action is required.

ADB7231W KEY LABEL cannot be applied to TB:*schema.table-name.*

Explanation

A key label cannot be added to the specified table, because the table does not reside in a universal table space or a partitioned (non-UTS) table space.

schema.table-name
The qualified name of the table.

System action:
Processing continues.

User response:
If you want a key label added to the specified table, you must create it in a universal table space or a partitioned (non-UTS) table space.

ADB7232E Change to DSSIZE is not valid for this segmented table space because of Db2 restrictions.

Explanation:
Db2 Object Comparison Tool cannot change the DSSIZE, because changing the DSSIZE is not valid for a segmented table space.

System action:
Processing stops.

User response:
Change the target table space to be a LOB or universal table space so that the DSSIZE change can be applied.

ADB7350E *<insert1>* detected an ONCODE condition *<ONCODE_value>* in *<internal_routine>* at *<line_number>*.

Explanation:
Internal error caused in location in specified module.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB7351I When the source is from DDL and the target is from DB2 catalog and APPLCOMPAT is not specified on the source DDL, or the source is from DB2 catalog and the target also is from the DB2 catalog, then Object Compare will use the target SYSPACKAGE.APPLCOMPAT value.

Explanation:
The APPLCOMPAT value from the target Db2 catalog is to be used, because the Db2 Admin Tool cannot determine whether APPLICOMPAT was specified in the CREATE or ALTER statement or the default value from the APPLCOMPAT subsystem parameter was used.

System action:
Processing continues.

User response

No action is required.

ADB7352I When the source is from DDL and the target is from DDL and APPLCOMPAT option is not specified on the source DDL, or the source is from DB2 catalog and the target is from DDL, then the DB2 subsystem parameter APPLCOMPAT default value will be used when the changes are applied to the target system.

Explanation:
When the changes are applied to a target system, the APPLCOMPAT subsystem parameter value is to be used. No APPLCOMPAT clause is generated.

System action:
Processing continues.

User response

No action is required.

ADB7353W Table space *dbname.tsname* is not specified in the source for the comparison. Consider including this table space *dbname.tsname* in the source.

Explanation

The target objects are automatically selected. However, the scope of the comparison source is not a table space. Therefore, if the compare option **Suppress DROP of objects** is set to NO, objects that are included only in the target might be dropped.

dbname.tsname
The table space to consider adding to the source.

System action:
Processing continues.

User response:

If dropping these objects is not intended, add the specified table space to the source.

ADB7380E **Module *module_name* - Severe error. *program_name* has been stopped.**

Explanation:
The Object Comparison tool has issued an error message for a severe problem.

System action:
A return code of 12 is set and processing stops.

User response:
Refer to other error messages generated in the same report for more information on the cause of this error and actions you can take.

ADB7385W **The Preserve all data (preserve_all_data) option was not activated for this compare process. Changes to whether a column can contain null values will be processed by using ALTER statements, which may result in data loss.**

Explanation:
Because the **Preserve all data** option or preserve_all_data CM batch parameter is set to NO, this change will result in data loss for the affected columns.

System action:
Processing continues.

User response:
If you want to save a copy of the altered data, set the **Unload altered tables** option or unload_altered_tables parameter to YES.

Related information

"PRESERVE_ALL_DATA" on page 709

ADB7386E **The PARM_COUNT for NSP *procedure-name* in the target cannot be altered to match the source, because version *version* does not exist in the source.**

Explanation

The specified native SQL procedure (NSP) on the target has a different number of parameters (PARM_COUNT in SYSIBM.SYSROUTINES) than the same NSP on the source. Object Comparison Tool cannot alter the number of parameters, because the indicated version of the target NSP does not exist on the source. (Because Object Comparison Tool cannot guarantee that the procedure bodies are the same, it does not alter procedure definitions when the versions for the source and target do not match.) Additionally, because the **Suppress DROP of objects** field (the KEEPTGT parameter in the JCL) was set to YES, the NSP on the target cannot be dropped. Therefore, processing halted with this error.

procedure-name
The procedure name

version
The version identifier

System action:
Processing stops.

User response

Take one of the following actions:

- Change the **Suppress DROP of objects** field (KEEPTGT parameter) to NO. In this case, the NSP version *version* on the target will be dropped.
- Provide a corresponding NSP on the source with the same name and version as the target.

ADB7387E **Index references dropped column *column_name***

Explanation

The index references a column that was dropped from the table. Therefore, the index cannot be created or recreated.

column_name
The referenced column that was dropped.

System action:
Processing stops.

User response:
Either drop the index or drop and recreate the index without the dropped column.

ADB7388W ***object-name* has been masked as *new-object-name* during this compare run.**

Explanation

The object name is masked for this comparison.

object-name
The original object name.

new-object-name
The object name with the mask applied.

System action:
Processing continues.

User response

No action is required.

ADB7389W **The source and target have different *bind-option* values. The target value will not be altered.**

Explanation

Object Comparison Tool detected different values for the indicated bind option in the source and target trigger packages. The target value will not be changed to match the source, because that change requires a rebind operation.

bind-option
The bind option that differed. Possible values are: SYSTEM_TIME SENSITIVE, BUSINESS_TIME SENSITIVE, or ARCHIVE SENSITIVE.

System action:
Processing continues.

User response

If you want the target to use the source value, rebind the target trigger package with the desired value by using the DB2I panels. Otherwise, no action is required.

ADB7390I **LOB table spaces will be converted from explicit to implicit table spaces when the associated base table is converted from non-UTS to UTS.**

Explanation:
For tables that contain LOB columns, Db2 requires that LOB table spaces and auxiliary tables be created to hold the LOB data. When the target base table in a non-UTS is compared to the source base table in a UTS, regardless of whether the source auxiliary tables are created explicitly, auxiliary tables are converted to implicit auxiliary tables so that auxiliary tables can be managed by Db2.

System action:
Processing continues.

User response

No action is required.

ADB7391E **The table access control option *value* is not supported. The table will not be added.**

Explanation

Because the indicated value is not a valid access control option, the table cannot be added. Acceptable values are listed in the description of the CONTROL column of the SYSTABLES catalog table.

value
The invalid SYSTABLES.CONTROL value.

System action:
Processing continues.

User response:
Correct the SYSTABLES.CONTROL value for the source table and run the comparison again.

ADB7392E **Option COMPRESS on table space *database.table-space-name* is changed to YES HUFFMAN. COMPRESS YES HUFFMAN is valid only for Db2 12 function level 509 and higher.**

Explanation

The COMPRESS YES HUFFMAN attribute is not valid for the target table space, because it is running on a Db2 version and function level lower than Db2 12 function level 509.

database.table-space-name
The qualified name of the table space.

System action:
Processing stops.

User response:
Remove the HUFFMAN attribute from the source table space or specify an ignore for this attribute and run the comparison again.

Related concepts

"Ignores" on page 846
An *ignore* provides the ability to specify that certain fields in the Db2 catalog records are to be ignored when objects are compared. Ignores help avoid meaningless comparisons and protect those fields, called *ignore fields*, from being changed.

ADB7393W **This trigger is dependent on view *view-name*. If this view is dropped and recreated, those changes will cascade to this trigger.**

Explanation

Because this trigger is dependent on a view, and Object Comparison Tool processes triggers before views, the trigger might still need to be dropped and recreated. At the time that this message is issued, Object Comparison Tool does not yet know the changes to the view. If the view is dropped and recreated (as reported in a later message), those changes will cascade to the trigger, and the trigger will also be dropped and recreated.

view-name
The qualified name of the view.

System action:
Processing continues.

User response

No action is required.

ADB7394E A table cannot be added to tablespace *database_name.tablespace_name*, because it is a UTS and already contains a table.

Explanation

A universal table space (UTS) can contain only one table. Because the specified UTS contains a table, Object Comparison Tool cannot create a new table in it.

database_name
The name of the database.

tablespace_name
The name of the table space.

System action:
Processing stops.

User response:
Create this table in a different table space or drop the existing table in the UTS.

ADB7395W Not enough information is available for tablespace *database_name.tablespace_name*. Make sure the tablespace does not already contain a table.

Explanation

Object Comparison Tool generated a statement to create a table in the specified universal table space (UTS). However, because the table space is not in the scope of the compare, Object Comparison Tool cannot determine whether the table space already contains a table. A UTS can contain only one table.

database_name
The name of the database.

tablespace_name
The name of the table space.

System action:
Processing continues.

User response:
Before running the CREATE TABLE statement, ensure that the specified UTS does not already contain a table.

ADB7397E The number of partitions exceeds the limit of 4096 for PAGENUM RELATIVE table spaces. Ensure that the number of partitions in the source does not exceed this maximum value.

Explanation:
Db2 limits the number of partitions to 4096 for table spaces with the PAGENUM RELATIVE attribute.

System action:
Processing stops.

User response:
When adding partitions, ensure that resulting number of partitions is less than or equal to 4096.

ADB7399E *module-name* The auxiliary table *AUX-table-name* cannot be processed without its base table *base-table-name* in the source.

Explanation

The specified auxiliary table cannot be processed in the target without the related base table. The base table cannot be processed, because the source DDL does not include the base table.

module-name
The name of the module.

AUX-table-name
The qualified name of the auxiliary table.

base-table-name
The qualified name of the base table.

System action:
Processing stops.

User response:
Include the related base table in the source DDL.

ADB7400I Object compare prepends SYSTEM PATH names SYSIBM, SYSFUN, SYSPROC, and SYSIBMADM in the SET PATH statement of the compare source if SYSTEM PATH names are not explicitly specified in the compare source.

Explanation:
Db2 Object Comparison Tool detected a PATHSCHEMAS change, and therefore added the listed SYSTEM PATH names. Db2 11 and later uses these SYSTEM PATH names for an unqualified PATH value. Thus, prepending these SYSTEM PATH names does not alter the PATH value that you specified in the compare source.

System action:
Processing continues.

User response

No action is required.

ADB7401E Compressed catalog record failed to decompress.

Explanation:
An error occurred while decompressing the compressed catalog record.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB7402E Unexpected record type found on *<insert1>* file. Expected: *<insert2>*. Found: *<insert3>*.

Explanation:
An unexpected record type has been found on source file or target file

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB7403E *<insert1>* limit reached. Max = *<insert2>*.

Explanation:
An error occurred when the number of elements in an array created for relations or user-defined functions reached the maximum limit.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB7404E Number of version file records generated for an object exceeds the limit.

Explanation:
Too many version file records have been generated for an object.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB7405E Duplicate drop is detected for object *<insert1>*.

Explanation:
A duplicate explicit drop was detected for an object.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB7406E Sort Process failed for *<insert1>* version file.

Explanation:
An error has occurred during the sorting process of source or target version file records.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB7407E Unknown catalog record type *<insert1>*.

Explanation:
An unknown record type has been found in the version file.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB7408E Server error when generating DDL.

Explanation:
An error occurred while generating DDL.

System action:
Processing stops.

User response:
Contact IBM Software Support.

ADB7409E UNLOAD cannot be generated for table *table_name* because of an incomplete state. *reason*

Explanation

An UNLOAD statement needs to be generated for a table to apply changes, but the table is in an incomplete state. UNLOAD is not allowed for tables in an incomplete state, so changes cannot be applied.

table_name
The name of the table in an incomplete state.

reason
One or more reasons for the incomplete state.

System action:
Processing stops.

Programmer response:
Correct the table to resolve the incomplete state and try again.

ADB7410W Option COMPRESS on table space *database.table-space-name* is changed to YES FIXEDLENGTH. COMPRESS YES FIXEDLENGTH is valid only for Db2 12 function level

509 and higher. Substituting with equivalent YES for this compare.

Explanation

The COMPRESS YES FIXEDLENGTH attribute is not valid for the target table space, because it is running on a Db2 version and function level lower than Db2 12 function level 509. For earlier versions and function levels, the equivalent attribute value is COMPRESS YES. Therefore, COMPRESS YES will be used for the target table space.

database.table-space-name
The qualified name of the table space.

System action:
Processing continues.

User response

No action is required.

ADB7411E ***schema.name* of base table for auxiliary table does not match that of the existing base table record.**

Explanation

A schema mismatch was found. The schema of the base table does not match the one in the version file for the auxiliary table.

schema.name
The qualified name of the base table.

System action:
Processing stops.

User response:
To ensure that the auxiliary table is linked with the correct base table, add the SET CURRENT SCHEMA statement to the source DDL.

ADB7701E **The DB2 Admin Tool was unable to load the DB2 DECP module. The return code is *rc*. The DDL reader was terminated.**

Explanation:
An internal error occurred while loading the DECP module.

System action:
A return code was set, and the DDL reader was terminated.

User response:
Correct the job by specifying a valid DECP loading action, and resubmit the job. If the problem persists, report this error to IBM Software Support.

ADB7705W **The DDL reader does not support the *statement_name* statement.**

Explanation:
An unsupported statement is specified in the DDL.

System action:
Processing continues with the next statement.

User response:
Fix the DDL by correcting the unsupported SQL statement and try again.

ADB7706E **The DDL reader does not support the *statement_name* statement.**

Explanation:
An unsupported statement is specified in the DDL.

System action:
Processing continues with the next statement.

User response:
Fix the DDL by correcting the unsupported SQL statement and try again.

ADB7708E ***attribute* for table *table-name* is invalid. The DDL statement is not processed.**

Explanation

An invalid attribute value was specified for the listed table.

attribute
The attribute name, such as BUFFERPOOL, MEMBER CLUSTER, or TRACKMOD.

table-name
The qualified name of the table for which the attribute was specified.

System action:
Processing continues.

User response:
Fix the DDL by correcting the attribute value.

ADB7709E **An error occurred while processing the SET PATH statement.**

Explanation:
The DDL reader encountered an invalid token in the SET PATH statement.

System action:
A return code of 12 is set and processing stops.

User response:
Specify a valid SET PATH statement and try again.

ADB7710E ***<attribute>* and *<attribute2>* were specified to *<objname>*. The statement cannot be executed.**

Explanation:
Mutually exclusive options were specified.

System action:

Processing continues with the next statement.

User response:
Fix the DDL by correcting the unsupported SQL statement and try again.

ADB7711I The DDL reader is processing under the authorization ID for the *userid* user ID. The authorization ID can be changed by the SET CURRENT SQLID statement.

Explanation:
The SQL ID is for informational purposes only.

System action:
The DDL reader continues processing.

User response:
No action is required.

ADB7713I The DDL reader is processing under the authorization ID for the *schema_name* schema. The authorization ID can be changed by the SET CURRENT SQLID statement.

Explanation:
The schema name is provided for informational purposes only.

System action:
The DDL reader continues processing.

User response:
No action is required.

ADB7754E The comment for the column *colname* in the table *name* is too long. Comments are limited to 762 characters.

Explanation:
The comment exceeds 762 characters.

System action:
Return code 8 is set, and processing continues.

User response:
Ensure that the length of the comment is less than or equal to 762 characters, and resubmit the job.

ADB7755E *<colname>* is not a column of table *<creator>*. *<name>*.

Explanation:
This error message is displayed if an invalid column name is specified.

System action:
A return code of 12 is set, and DTC continues processing.

User response:
Fix the problem and try again.

ADB7757E The following element that is specified for an IDENTITY column is not valid: *token_name*.

Explanation:
While processing an AS IDENTITY clause, the DDL reader encountered an invalid token in the data type expression.

System action:
This SQL statement cannot be executed. A return code of 16 is set and processing stops.

User response:
Correct the DDL with valid statements, and resubmit the job.

ADB7763I Temporary (TEMP) databases are not supported by DB2 V9 or higher.

Explanation:
The DDL has a TEMP DATABASES statement, which is not supported by DB2 9or higher.

System action:
The DDL reader continues processing.

User response:
No action is required.

ADB7715E The DDL reader could not parse a DDL statement. The return code is *rc*. The error statement is *error_stmt*.

Explanation:
The specified statement, *error_stmt*, is an invalid SQL statement.

System action:
The DDL reader stops processing.

User response:
Correct the DDL with valid a SQL statement, and resubmit the job.

ADB7719W No action taken for the GRANT on *type* statement. Processing continues with the next statement.

Explanation:
A GRANT statement specified in either a package or a plan was ignored.

System action:
The DDL reader continues processing the next statement.

User response:
No action is required.

ADB7723E There is an unexpected token in the table definition. The token is *token_name*.

Explanation:
The token in the table definition is not valid. The SQL statement cannot be executed.

System action:
A return code of 16 is set and processing stops.

User response:
Correct the DDL and resubmit the job.

ADB7725E There is an unexpected token in the table definition. The token is *token_name*.

Explanation:
The token in the table definition is not valid. The SQL statement cannot be executed.

System action:
A return code of 16 is set and processing stops.

User response:
Correct the DDL and resubmit the job.

ADB7727E An unexpected token was found in the parameter declaration. The token is *token_name*.

Explanation:
The specified *token_name* is not a valid UDF parameter name. The SQL statement cannot be executed.

System action:
A return code of 16 is set and processing stops.

User response:
Correct the DDL and resubmit the job.

ADB7729E An unexpected token was found in the RETURNS clause. The token is *token_name*.

Explanation:
The specified *token_name* is not a valid UDF parameter name. The SQL statement cannot be executed.

System action:
A return code of 16 is set and processing stops.

User response:
Correct the DDL and resubmit the job.

ADB7731W A function option was specified that is not valid. The *token_name* token was found in the following position: *processing_position*.

Explanation:
The DDL reader encountered a token where it expected to find a function option.

System action:
The DDL reader continues processing.

User response:
Correct the option name and resubmit the job.

ADB7733E A function option was specified that is not valid. The *token_name* token index is out of range.

Explanation:
The DDL reader encountered an invalid *token_name* as a function option. The SQL statement cannot be executed.

System action:
The DDL reader continues processing.

User response:
Correct the DDL and resubmit the job. If the problem persists, report this error to IBM Software Support.

ADB7735E The following element that is specified in an ALTER SEQUENCE statement is not valid: *token_name*.

Explanation:
While processing an ALTER statement, the DDL reader encountered an invalid token in the SEQUENCE expression. The SQL statement cannot be executed.

System action:
A return code of 16 is set and processing stops.

User response:
Correct the DDL and resubmit the job.

ADB7739E The DDL reader encountered nested bracketed comments that are not complete.

Explanation:
The DDL reader encountered nested brackets that are not matched. The SQL statement cannot be executed.

System action:
A return code of 8 is set and processing stops.

User response:
Ensure every comment has an opening and closing bracket. Correct the DDL and resubmit the job.

ADB7741E The number of entries in the Token index exceeds the number of tokens.

Explanation:
The DDL reader encountered a mismatch between the token index number and the number of tokens. The SQL statement cannot be executed. This is an internal error.

System action:
A return code of 16 is set and processing stops.

User response:
Report this error to IBM Software Support.

ADB7743E The DDL reader encountered an SQL statement that is too long.

Explanation:
This is an internal error. The SQL statement cannot be executed.

System action:
A return code of 16 is set and processing stops.

User response:
Correct the DDL with valid statements, and resubmit the job. If the problem persists, report this error to IBM Software Support.

ADB7753E The following string is too long: *string*.

Explanation:
The DDL reader encountered a quoted string that is too long. This is an internal error.

System action:
A return code of 12 is set and processing stops.

User response:
Report this error to IBM Software Support.

ADB7749E The DDL reader encountered an unexpected token in the following option: *option_name*.

Explanation:
The specified *option_name* is not a valid option name in the SQL PROCEDURE statement. This SQL statement cannot be executed.

System action:
A return code of 16 is set and processing stops.

User response:
Correct the option name and resubmit the job.

ADB7751I The *token_name* token was found in the Column list, but it is not valid.

Explanation:
While creating a table, the DDL reader encountered an invalid token in the Column list.

System action:
The DDL reader continues processing.

User response:
Specify valid SQL statements in the DDL and try again.

ADB7756W The following user-defined type (UDT) is not defined: *typename*.

Explanation:
This error message is displayed if the UDT name is not defined in an input file along with its referencing object. If the UDT definition is missing, the downstream process may result in an error.

System action:
A return code of 4 is set and processing continues.

User response:
Define the UDT name and submit the job again if necessary.

ADB7765E The DDL reader encountered the following invalid token after an IN clause: *token_name*.

Explanation:
While creating a table, the DDL reader encountered an invalid token.

System action:
A return code of 16 is set and processing stops.

User response:
Fix the DDL with valid SQL statements and try again.

ADB7767I The DDL reader encountered the following partition number, which is not valid: *partition*.

Explanation:
The DDL reader encountered a partition number that exceeds the number of partitions in the table space.

System action:
The DDL reader continues processing.

User response:
Specify the valid partition number in the DDL and try again.

ADB7769E Empty parentheses () are not permitted following the FLOAT keyword.

Explanation:
The DDL FLOAT keyword needs a numerical expression inside parentheses in order for the floating point expression to be translated.

System action:
A return code of 16 is set and processing stops.

User response:
Provide a numeric expression in the FLOAT keyword and try again.

ADB7771E The DDL reader encountered a substring outside of a string.

Explanation:
This is an internal error caused by an invalid string position.

System action:
This SQL statement cannot be executed and processing stops.

User response:

Correct the DDL with valid statements, and resubmit the job. If the problem persists, report this error to IBM Software Support.

ADB7773W **No action was taken for the ALTER *type* REGENERATE statement. Processing continues with the next statement.**

Explanation:
The DDL reader encountered an unsupported ALTER *type* REGENERATE statement such as INDEX, MASK, PERMISSION, and PROCEDURE.

System action:
The DDL readers continues processing.

User response:
Fix the DDL with supported SQL statements and try again.

ADB7775I **The DDL reader does not support the *type* statement. Processing continues with the next statement.**

Explanation:
A ROLE or TRUSTED context is specified in an SQL statement, which is not supported.

System action:
The DDL readers continues processing.

User response:
Fix the DDL with supported SQL statements and try again.

ADB7776E **The length specification of the column *col_name* in table *table_name* is invalid.**

Explanation:
The length specification of the column is invalid.

System action:
Processing stops.

User response:
Fix the DDL with supported SQL statements and try again.

ADB7777E **The DDL reader encountered a CCSID *ccsid|clause* which is not valid for the *encoding|data* type *type*.**

Explanation:
The DDL reader encountered a CCSID that is not valid for the data type or encoding type. The DDL reader uses the DB2 encoding scheme to verify CCSID values.

System action:
None. DTC continues processing.

User response:
Fix the DDL with a valid CCSID, and try again.

ADB7778E **The DDL reader encountered mutually exclusive clauses in the ALTER TABLE statement.**

Explanation:
You can only specify the same clause once, except for the ADD COLUMN and ALTER COLUMN clauses. The ALTER COLUMN, ADD PARTITION, and ROTATE PARTITION clauses are mutually exclusive.

System action:
The DDL reader stops processing.

User response:
Fix the DDL with supported SQL statements and try the operation again.

ADB7830E **The node with the key *key_name* already exists in the dictionary.**

Explanation:
The node with the key cannot be inserted into the dictionary because the key already exists. This is an internal error.

System action:
Processing stops.

User response:
Report this internal error to IBM Software Support.

ADB7900I **Version File is at current level: *version_level*. No conversion necessary.**

Explanation:
The Version File does not need to be converted.

System action:
Return code = 0. Processing continues.

User response:
No action is required.

ADB7902E **Unsupported DB2 release: *DB2 release***

Explanation:
The DB2 release from the Version File header is not supported.

System action:
Return code = 12. Processing stops.

User response:
Recreate a new version file at the current level and then try again.

ADB7904E **Unsupported or invalid version file row type: *row type***

Explanation:
A version file row type is not valid.

System action:
Return code = 8. Processing continues.

User response:
This is a processing error. Contact IBM Software Support.

ADB7910E Version File error: *error code*

Explanation

A Version File error has occurred. The type of error is indicated by the errorcode.

Errorcode = 8: The version file is empty.

Errorcode = 12: The version file is missing or its name is not correct.

System action:
The Version File Conversion Tool terminates processing.

User response:
Correct the data set name or member name.

ADB7913E The old and new version files have the same name: *name*

Explanation:
The new version file and the old version file have the same name.

System action:
A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response:
Correct the new version file dataset name or member name.

ADB7915E Invalid combination of parameters.

Explanation:
The following combinations of keyword parameters are valid: VFOLD and VFNEW, VOWNER and VNAME, or VID.

System action:
A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response:
Correct the parameters.

ADB7916E Parameter *parameter* specfied, *missing parm* omitted. Both are required. The version file is not defined.

Explanation:
You must specify both VFOLD and VFNEW parameters or both VOWNER and VNAME parameters for the Version File conversion tool.

System action:
A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response:
Correct the parameters.

ADB7917E Unable to access *parameter* version file *version_file*

Explanation:
The specified version file could not be opened. Correct the file name.

System action:
A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response:
Correct the parameters.

ADB7918E Version id = *version_ID* was not found.

Explanation:
The specified version ID was not found in the base version database.

System action:
A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response:
Correct the parameters.

ADB7919E Unable to determine the DB2 version for row type: *row_type*.

Explanation:
The DB2 version for the specified version file row type could not be determined.

System action:
A return code of 12 is set. The Version File Conversion Tool terminates processing the current version file.

User response:
Contact IBM Software Support.

ADB7923E The keyword parameter, *keyword* is invalid.

Explanation:
The specified keyword parameter is invalid for the Version File Conversion Tool.

System action:
A return code of 12 is set. The Version File Conversion Tool terminates processing the current version file.

User response:
Correct the keyword parameter and try again.

ADB7928E The version file has an unknown version level.

Explanation:
An unknown version was detected in the version file. The version file was created by a prior release.

System action:

A return code of 8 is set. The Version File Conversion Tool terminates processing the current version file.

User response:
Recreate a new version file at the current level and then retry.

ADB7951E **An invalid action was specified for the saved compare results.**

Explanation:
The input job that Object Compare generated contains an invalid value for the action to save the compare results. The action for the saved compare results must be either ADD or REPLACE.

System action:
A return code of 12 is set, and processing stops.

User response:
Edit the input job to specify a valid action, and resubmit the job. Report this internal error to IBM Software Support.

ADB7952W **IBMDB2 Analytics Accelerator is not available for the current DB2 subsystem.**

Explanation:
Admin or Object Comparison Tool has detected and reloaded the accelerated tables that contain modified data, but DB2 Analytics Accelerator is not available for the current DB2 subsystem.

System action:
A return code of 4 is set, and processing continues.

User response:
Turn off Reload accelerated tables on panel ADB2PCO or install DB2 Analytics Accelerator for the current DB2 subsystem.

ADB7953I ***module-name***

Explanation

This informational message introduces report output. The content of the message varies depending on the context.

module-name
The module name is listed when this message is included in the ADBMSGS data set.

System action:
Processing continues.

User response:
None.

Related tasks

“Consolidating messages into a single file” on page 246
Some batch interface jobs produce multiple output files, each of which contain a number of messages. To simplify troubleshooting, you can consolidate these messages into one file, or data set. You can do this consolidation for batch jobs for functions such as Change Management (CM) batch interface, ALT, Compare, and ADBTEP2.

ADB7954W ***module-name***

Explanation

This message is a generic warning message. The content of the message varies depending on the context.

module-name
The module name is listed when this message is included in the ADBMSGS data set.

System action:
Processing continues.

User response:
Verify whether this message is expected and make changes as needed.

Related tasks

“Consolidating messages into a single file” on page 246
Some batch interface jobs produce multiple output files, each of which contain a number of messages. To simplify troubleshooting, you can consolidate these messages into one file, or data set. You can do this consolidation for batch jobs for functions such as Change Management (CM) batch interface, ALT, Compare, and ADBTEP2.

ADB7955E ***module-name***

Explanation

This message is a generic error message. The content of the message varies depending on the context.

module-name
The module name is listed when this message is included in the ADBMSGS data set.

System action:
Processing fails and stops.

User response:
Fix the problem and try again.

Related tasks

“Consolidating messages into a single file” on page 246
Some batch interface jobs produce multiple output files, each of which contain a number of messages. To simplify troubleshooting, you can consolidate these messages into one file, or data set. You can do this consolidation for batch

jobs for functions such as Change Management (CM) batch interface, ALT, Compare, and ADBTEP2.

ADB7956I ***module-name***

Explanation

This informational message indicates the beginning and end of sections of output. The content of the message varies depending on the context.

module-name
The module name is listed when this message is included in the ADBMSGS data set.

System action:
Processing continues.

User response:
None.

Related tasks

"Consolidating messages into a single file" on page 246
Some batch interface jobs produce multiple output files, each of which contain a number of messages. To simplify troubleshooting, you can consolidate these messages into one file, or data set. You can do this consolidation for batch jobs for functions such as Change Management (CM) batch interface, ALT, Compare, and ADBTEP2.

ADB7957I ***module-name* CM batch parameters - *list-type*:**

Explanation

This informational message introduces a list of Change Management (CM) parameter values in the ADBMSGS data set.

module-name
The name of the module that issued the message.

list-type
The type of list. *list-type* can have one of the following values:

init and PROF
The list reports the initial and profile values for the parameters.

PARMS DD applied
The list reports the parameter values that were provided in an invocation override (by using the PARMS DD statement).

final
The list reports the final parameter values that were used.

Within each list, the parameters are ordered alphabetically. CM batch parameters with blank values are not listed.

System action:
Processing continues.

User response

No action is required.

Related tasks

"Consolidating messages into a single file" on page 246
Some batch interface jobs produce multiple output files, each of which contain a number of messages. To simplify troubleshooting, you can consolidate these messages into one file, or data set. You can do this consolidation for batch jobs for functions such as Change Management (CM) batch interface, ALT, Compare, and ADBTEP2.

ADB8001E **The second record in a record pair was not found in the input version file.**

Explanation:
During the merge operation, required information was not available.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8002E **An internal error occurred for an unknown row type of *<type>*.**

Explanation:
The input version file format is not valid.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8003E **A wildcard character (*) was used to specify a volume name, but an explicit volume name has already been specified.**

Explanation:
An error occurred while an ALTER storage group was being processed.

System action:

Processing stops.

User response:
Review the volume name, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8004E **The volume ID, *volume_id*, cannot be added because a wildcard character (*) was already specified on the storage group.**

Explanation:
An error occurred while an ALTER storage group was being processed.

System action:
Processing stops.

User response:
Review the volume ID, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8005E **The volume ID, *volume_id*, cannot be added to storage group, *obj_name*. The volume is already part of the storage group.**

Explanation:
An error occurred while an ALTER storage group was being processed.

System action:
Processing stops.

User response:
Review the volume ID, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8006W **An attempt was made to drop an *obj_type obj_name* that does not exist.**

Explanation:
During the merge operation, an error occurred and the object could not be dropped.

System action:
Processing continues.

User response:
If necessary, ensure that the object to be dropped is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8007E **An attempt was made to create an *obj_type* that already exists.**

Explanation:
During the merge operation, an error occurred and the object was not created.

System action:
Processing stops.

User response:
Ensure that the object to be created is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8008E **An internal error occurred for an unknown transaction.**

Explanation:
An error occurred while an object was being processed.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8009E **An attempt was made to rename an *obj_type* that does not exist.**

Explanation:
During the merge operation, an error occurred and the object could not be found and renamed.

System action:
Processing stops.

User response:
Ensure that the object to be renamed is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8010E **An object cannot be renamed to a specified new name because the new name was already specified in a previous rename operation.**

Explanation:
An attempt was made to rename an object. The new name was assigned in a previous rename operation and cannot be used for this object.

System action:
Processing stops.

User response:
Ensure that the rename that was specified is unique and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8011E **An attempt was made to alter an object that does not contain the record to change.**

Explanation:
During the merge operation, an error occurred. No object row was found to match a delta row of a specific type.

System action:
Processing stops.

User response:
Ensure that the object, and particularly the row type, is specified correctly. Correct the appropriate statements and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8012E **An expected version file record *row_type* was not found in a base version record.**

Explanation:
During the merge operation, an error occurred. A record of a specific row type was expected but was not found.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8013E **An attempt was made to alter a column record, but the specified table does not contain this column *column_name*.**

Explanation:
During the merge operation, an error occurred. A column, specified to be updated when altering a table, was not found.

System action:
Processing stops.

User response:
Ensure that the object, and particularly changes to the column records, is specified correctly. Correct the appropriate statements and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8014E **An attempt was made to alter the table attributes of a table that does not contain the *rowtype* record to be changed.**

Explanation:
During the merge operation, an error occurred. A column record, of a specific row type and specified to be updated when altering a table, was not found.

System action:
Processing stops.

User response:
Ensure that the object, and particularly the row type, is specified correctly. Correct the appropriate statements and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8015E **An attempt was made to change the access control for a table that cannot be found.**

Explanation:
An error occurred while access to a table row or column was being activated or deactivated.

System action:
Processing stops.

User response:
Review the access control specified for the table, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8063I.

ADB8016E **An attempt was made to add or alter the ORGANIZE BY HASH clause for a table, but the corresponding record in the table was not found.**

Explanation:
During the merge operation, an error occurred. The ORGANIZE BY HASH clause could not be used in a table object.

System action:
Processing stops.

User response:
Ensure that the table object is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8017E **An attempt was made to add a SYSTEM_TIME or BUSINESS_PERIOD clause to a table, but the corresponding record in the table was not found.**

Explanation:
During the merge operation, an error occurred. A row that was specified to be updated when altering a table was not found.

System action:
Processing stops.

User response:
Ensure that the table object is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8018E **An attempt was made to add the *col_name* column to the**

***table_name* table, but *col_name* already exists in this table.**

Explanation:
During the merge operation, an error occurred and the column was not added.

System action:
Processing stops.

User response:
Ensure that the object to be created is specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8019E **An attempt was made to drop an *obj_type* that does not exist.**

Explanation:
During the merge operation, an error occurred and the object could not be dropped.

System action:
Processing stops.

User response:
Ensure that the object to be dropped is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8020E **An attempt was made to add a primary key to the table *table_name* , but this table already has a primary key.**

Explanation:
An error occurred while a primary key was being added to a table. A table can have only one primary key.

System action:
Processing stops.

User response:
Ensure that the table and the key to be added are specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8021E **An attempt was made to add a primary or unique constraint to the *table_name* table, but a constraint with the same name already exists for this table.**

Explanation:
An error occurred while a primary or unique constraint was being added to a table.

System action:
Processing stops.

User response:
Ensure that the table and the constraint to be added are specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8022E **An attempt was made to add a primary or unique key, but the column associated with the key, *col_name* was not found.**

Explanation:
An error occurred while a primary or unique key was being added to a table.

System action:
Processing stops.

User response:
Ensure that the table column and the key to be added are specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8023E **An attempt was made to add the table check constraint *check_name* to the table *table_name*, but a constraint with the same name already exists for this table.**

Explanation:
An error occurred while a table check constraint was being added a table. The same constraint name is already being used as a different check.

System action:
Processing stops.

User response:
Ensure that the table and the table check constraint to be added are specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8024E **The merge process stopped due to severe errors.**

Explanation:
The merge process stopped due to severe errors.

System action:
Processing stops.

User response:
Review other messages that accompany this message to determine the appropriate response.

ADB8025E **An attempt was made to process an invalid add operation for a table.**

Explanation:

An internal error occurred while processing an ADD operation for a table. The operation type is not valid.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8026E **The ROTATE PARTITION option cannot be processed. Reason code = rc.**

Explanation

An error occurred while an ALTER TABLE statement that specifies rotating partitions was being processed. The reason code indicates the source of the error:

1
The table is not partitioned

2
No table partitions exist

3
The row specified for rotate is unknown.

System action:
Processing stops.

User response:
Review the ALTER TABLE statement that was specified, particularly the ROTATE PARTITION option. Correct the appropriate statements and try again. If the reason code is 3, contact IBM Software Support and provide the information in this message.

ADB8027E **An error occurred in the ADBDICT module: msg.**

Explanation:
An internal error occurred in a dictionary module.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8028E **An attempt was made to drop a column, but that column does not exist in the *obj_type*.**

Explanation:
During the merge operation, an error occurred and the column was not removed from the object.

System action:
Processing stops.

User response:
Ensure that the object to be dropped is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8029E **An attempt was made to drop the check constraint *const_name*, but that constraint does not exist in the table *table_name*.**

Explanation:
An error occurred while a constraint was being dropped from a table.

System action:
Processing stops.

User response:
Ensure that the table and the constraint to be dropped are specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8030E **An attempt was made to process an invalid drop operation for a table.**

Explanation:
An internal error occurred while processing a DROP operation for a table.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8031E **An attempt was made to insert a column *col_name* in a table *table_name*, but *col_name* already exists in this table.**

Explanation:
An error occurred while a column was being inserted into a table. The column already exists.

System action:
Processing stops.

User response:
Ensure that the object to be inserted is specified correctly, correct the relevant statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8032E **An attempt was made to insert a column *col_name*, but the specified position was not found.**

Explanation:

An error occurred while a column was being inserted into a table. During the merge operation, the position for column was determined to be invalid.

System action:
Processing stops.

User response:
Ensure that the column is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8057I.

ADB8033E **An internal error occurred. The table was not in the dictionary.**

Explanation:
An internal error occurred during the renaming of a table.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8034E **COMMENT ON or LABEL ON on a column for the VIEW *obj_name* cannot be processed. Column *col_name* is not in the view.**

Explanation:
The comment or label on a statement is ignored because the column was not found in the view.

System action:
Processing continues.

User response:
Ensure that the column is specified correctly, correct the appropriate statements, and run the job again.

ADB8035E **An attempt was made to update a version file row, but the matching row specified in a delta change was not found.**

Explanation:
An internal error occurred while an object was being altered.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8063I.

ADB8036E **MERGE could not find and update the *obj_type* version.**

Explanation:
An internal error occurred while processing an ALTER FUNCTION or ALTER PROCEDURE statement. During the merge operation, the version of the stored procedure or function was not found.

System action:
Processing stops.

User response:
Refer to message ADB8057I to determine the stored procedure or function that could not be found and then review the specified ALTER FUNCTION or ALTER PROCEDURE statement that was specified.

ADB8037E **An attempt was made to replace a function with version *ver_id*, but that version does not exist.**

Explanation:
An error occurred while processing an ALTER function statement. During the merge operation, the specified version of the function was not found.

System action:
Processing stops.

User response:
Refer to message ADB8057I to determine the specific function and then review the specified ALTER FUNCTION statement.

ADB8038I **A DDL statement could not be parsed. Processing continues. RC = *<return_code>*.**

Explanation

An error occurred while processing a DDL statement of an object. The reason code indicates the source of the error:

1
An error occurred for a view object.

2
An error occurred for a RENAME statement.

System action:
Processing continues.

User response:
Correct the DDL statement, if necessary, and run the job again.

ADB8039E **MERGE encountered an error while registering an object. The *obj_type* already exists with the same name of *obj_name*.**

Explanation:
An error occurred while an object was being registered. An object with the same object name already exists.

System action:

Processing stops.

User response:
Ensure that the object is specified correctly, correct the appropriate statements, and try again. Details about the object that caused the error are provided in message ADB8063I.

ADB8040E **An error occurred during sort processing of the *vf_type* file: Return code from SORT = *return_code*.**

Explanation:
An internal sort process resulted in an error.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support and provide the return code and the information in message ADB8057I.

ADB8041W **Dropped foreign key *key_name* for table *obj_name* does not exist. The foreign key might have been dropped when the parent key was dropped.**

Explanation:
The specified foreign key does not exist.

System action:
Processing continues.

User response:
If necessary, ensure the foreign key is specified correctly and run the job again.

ADB8042I **No records were found in the base version file.**

Explanation:
During the merge operation, no records were found in the base version file.

System action:
Processing continues.

User response:
Review the base version file. Correct the file, if necessary, and run the job again.

ADB8043I **No delta changes to process.**

Explanation:
No change records were found in the delta version file.

System action:
Processing continues.

User response:
Review the change and the delta version file. Correct the file, if necessary, and run the job again.

ADB8044I **No objects to process.**

Explanation:
No input records were found.

System action:
Processing continues.

User response:
Review the base and delta version files. Correct the files, if necessary, and run the job again.

ADB8045I **The number of catalog rows exceeds the limit specified for the process.**

Explanation:
The number of catalog rows exceeds the limit specified for the process.

System action:
Processing continues.

User response:
This is an internal error. If necessary, contact IBM Software Support.

ADB8046W **The volume *vol_id* that was specified to be removed was not found in the storage group *obj_name*.**

Explanation:
The volume ID to be removed was not found in the storage group.

System action:
Processing continues.

User response:
If necessary, locate the volume, confirm that removal was specified, and then run the job again.

ADB8047E **KY rows were not found. Alter was attempted for the implicit unique index for table *table_name*.**

Explanation:
: An internal error occurred during the altering of an implicit index for a table.

System action:
Processing stops.

User response:
This is an internal error. Contact IBM Software Support. Details about the object that caused the error are provided in message ADB8057I.

ADB8048E **COMMENT ON or LABEL ON on a column for the VIEW *obj_name* cannot be processed. Column *col_name* is not in the view.**

Explanation:
: The comment or label on a statement is ignored because the column was not found in the view.

System action:
Processing continues.

User response:
Ensure that the column is specified correctly, correct the appropriate statements, and run the job again.

ADB8049I **During the ALTER procedure, *obj_type obj_name* was found, but the *obj_type* was not found. The *obj_type* is assumed to be implicitly created.**

Explanation:
: An attempt was made to alter an implicitly created object. Implicitly created objects cannot be altered.

System action:
Processing continues.

User response:
No response.

ADB8050W **Drop alias *obj_name* ignored. Alias does not exists.**

Explanation:
: A Drop Alias statement is ignored because the alias does not exit.

System action:
Processing continues.

User response:
: If necessary, correct the change and run the job again.

ADB8051W **Alter found for *obj_name*, but no object definition was found in base.**

Explanation:
: A change was found for an object, but no base definition for the object was found.

System action:
Processing continues.

User response:
: Ensure that the object is specified correctly, correct the appropriate statements, and run the job again. Details about the object that caused the error are provided in message ADB8057I.

ADB8052E **A change was found for *obj_name*, but no object definition was found in base.**

Explanation:
: A delta change exists for an object that is not defined.

System action:
Processing stops.

User response:
: Ensure that the object is specified correctly, correct the appropriate statements, and run the job again. Details about the object that caused the error are provided in message ADB8057I.

ADB8053W **A drop was specified for *obj_name*, but no object definition was found in base.**

Explanation:
: An attempt was made to drop an object that is not defined.

System action:
Processing continues.

User response:
: Ensure that the object is specified correctly, correct the appropriate statements, and run the job again.

ADB8054I **Internal rows, AR or XR, were not found during a search of the LOB or XML entries in the base version file. Row type: *row_type*.**

Explanation:
: During the merge operation, an expected auxiliary table or XML record was not found in the base records.

System action:
Processing continues.

User response:
: Review LOB and XML entries. If necessary, correct the statements and run the job again.

ADB8055I **The row type CO was not found in delta stack. No match to the corresponding implicit rows in the base change was found for rowtype: *row_type*.**

Explanation

:

An implicit column change, which was flagged as a delta change, was found, but no matching column definition was found. The implicit rows might have been created during internal processing.

.

System action:
Processing continues.

User response:
: Ensure that the object, and particularly the row type, is specified correctly. If necessary, correct the appropriate statements and run the job again.

ADB8056E **The statement CREATE TABLE *<table>* LIKE *<table>* is not yet supported.**

Explanation:

: The statement CREATE TABLE with LIKE predicate is not supported in the merge operation.

System action:
Processing stops.

User response:
: Remove the statement and try again.

ADB8057I **An error occurred during MERGE processing. The following details apply to the error: Operation: *operation*, Object name: *object_name*, Row type: *row_type*, Procedure: *proc_name*.**

Explanation:
The message text provides details about objects and procedures that are involved in the error.

System action:
Processing continues.

User response:
Use the message text information to correct the problem, or provide the information when you contact IBM Software Support.

ADB8058W **The statement CREATE TABLE *<table>* LIKE *<table>* is not yet supported.**

Explanation:
: The statement CREATE TABLE with LIKE predicate is not supported in the merge operation. The statement is ignored.

System action:
Processing continues.

User response:
: Optionally, remove the statement and try again.

ADB8062E **No object row matches the delta row. The row type is *row_type*. Reason code is *reason_code*.**

Explanation

The row type specified in a delta change was not found. One of the following reason codes is issued:

1

Error while processing an implicit non-LOB, non-XML table space.

2

Error while processing ALTER INDEX attributes.

3

Error while processing implicit index attributes.

System action:
Processing stops.

User response:
If necessary, correct the DDL statement, and try again.

ADB8063E **The rearranged column list does not match table *table_name*. The *column_name* column is not in the table.**

Explanation:
The list of columns to be rearranged does not match the table.

System action:
Processing stops.

User response:
If necessary, correct the DDL statement, and try again.

ADB8064E **The *column_name* column to be added to the *index_name* index is already an index column.**

Explanation:
A column can't be added to an index that already exists.

System action:
Processing stops.

User response:
If necessary, correct the DDL statement, and try again.

ADB8065E **The row type is invalid for Alter Index *operation_name*.**

Explanation:
An invalid operation was found while the Alter Index statement was being processed.

System action:
Processing stops.

User response:
If necessary, correct the DDL statement, and try again.

ADB8066E **The foreign key *key_name* to be added to table *table_name* already exists.**

Explanation:
An existing foreign key cannot be added to a table.

System action:
Processing stops.

User response:
If necessary, correct the DDL statement, and try again.

ADB8067E **Alter inconsistency in the row type, length, or prefix.**

Explanation:
An internal error occurred while a version file row was being altered.

System action:
Processing stops.

User response:

Gather the details about the object that caused the error from the ADB8057I message, and contact IBM Software Support.

ADB8070E **Table space *table_space_name* cannot be dropped, because it contains a history table.**

Explanation

An attempt was made to drop the specified table space, which contains a history table. As part of the Db2 restrictions for system-period data versioning, you cannot drop a table space that contains a history table.

table_space_name
The qualified name of the table space.

System action:
Processing stops.

User response:
None required. However, if you still want to drop the specified table space, you must drop versioning first so that the table in the table space is no longer a history table.

Related reference

Restrictions for system-period data versioning (Db2 12 for z/OS)

ADB8071E **Versioning cannot be dropped from table *table_name***

Explanation

An attempt was made to drop versioning from the specified table, but this DROP operation is invalid.

table_name
The qualified name of the table.

System action:
Processing stops.

User response:
Read the restrictions for the DROP VERSIONING clause in ALTER TABLE (Db2 12 for z/OS), correct the DDL statement, and try again.

ADB8072E **A severe error occurred when processing the specified limit key for *object-type object-qualifier.object-name*. *reason-text***

Explanation

The limit key is not valid.

object-type
The type of object. Possible values are Table, Table space, or Index.

object-qualifier.object-name
The qualified name of the object.

reason-text
The reason that the limit key is not valid.

System action:
Processing stops.

User response:
Change the limit key to a valid value.

ADB8073E **The partition cannot be added or inserted, because a partition with limit key *limit-key* already exists.**

Explanation

Each partition must have a unique limit key value, and the specified limit key value is already one of the limit key values for the table.

limit-key
The specified limit key value.

System action:
Processing stops.

User response:
Specify a unique limit key value.

ADB8074E **Parent table space *name* not found. Please include CREATE TABLESPACE statement. Table space is required to create table. Table space may be implicitly dropped via a DROP TABLE statement and not recreated.**

Explanation

The table space in which the table is to be created was not found.

This missing table space could be due to a prior DROP TABLE operation. Beginning in Db2 12 function level 506, a DROP TABLE operation implicitly drops the parent table space.

name
The name of the table space.

System action:
Processing stops.

User response:
Add the CREATE TABLESPACE statement for the parent table space to the DDL.

ADB8998I ***module-name* ================= Start of Report ===============**

Explanation

This message indicates the beginning of the object comparison report in the ADBMSGS data set.

module-name
The module name.

System action:
Processing continues.

User response:
None.

Related tasks

"Consolidating messages into a single file" on page 246
Some batch interface jobs produce multiple output files, each of which contain a number of messages. To simplify troubleshooting, you can consolidate these messages into one file, or data set. You can do this consolidation for batch jobs for functions such as Change Management (CM) batch interface, ALT, Compare, and ADBTEP2.

ADB8999I ***module-name*** **================== End of Report ==================**

Explanation

This message indicates the end of the object comparison report in the ADBMSGS data set.

module-name
The module name.

System action:
Processing continues.

User response:
None.

Related tasks

"Consolidating messages into a single file" on page 246
Some batch interface jobs produce multiple output files, each of which contain a number of messages. To simplify troubleshooting, you can consolidate these messages into one file, or data set. You can do this consolidation for batch jobs for functions such as Change Management (CM) batch interface, ALT, Compare, and ADBTEP2.

ADB9001W **A parameter name in the input parameter file was not recognized.**

Explanation:
The input parameter file contains a parameter name that is not valid. The job might not have run correctly because of the incorrect parameter name.

System action:
A return code of 4 is set, and processing continues. The program continues on to the next request after reporting the error.

User response:
Correct the invalid parameter, and resubmit the job.

ADB9002W **Comments are not allowed in the input parameter file.**

Explanation:
The input parameter file cannot contain comments. The job might not have run correctly.

System action:
A return code of 4 is set, and processing continues. The program continues on to the next request after reporting the error.

User response:
Delete the comments from the input parameter file, and resubmit the job.

ADB9003W **Invalid input from the input parameter file is ignored.**

Explanation:
The input parameter file contains invalid input, which is ignored. The job might not have run correctly because of the invalid input.

System action:
A return code of 4 is set, and processing continues. The program continues on to the next request after reporting the error.

User response:
Correct the invalid parameter, and resubmit the job.

ADB9004W **Processing continues.**

Explanation:
This message is issued in conjunction with message ADB9001, ADB9002, or ADB9003 to indicate that processing continues when the program encounters these errors.

System action:
Processing continues.

User response:
None.

ADB9005W **The following input was skipped: *error_text*.**

Explanation:
The job might not have run correctly because input was skipped. *error_text* identifies the input that was skipped.

System action:
A return code of 4 is set, and processing continues.

User response:

Correct the input, and resubmit the job.

ADB9006I The program *program_name* completed abnormally.

Explanation:
The accompanying messages indicate why the identified program did not complete normally.

System action:
None..

User response:
See the accompanying messages in the report.

ADB9007E A version name was not specified.

Explanation:
The request cannot be processed because a version name was not specified.

System action:
A return code of 12 is set, and processing stops.

User response:
Specify a valid version name, and resubmit the request.

ADB9008E A version qualifier was not specified.

Explanation:
The request cannot be processed because a version qualifier was not specified.

System action:
A return code of 12 is set, and processing stops.

User response:
Specify a valid version qualifier and resubmit the request.

ADB9009E Package *module_name* needs to be bound or rebound.

Explanation:
An SQL statement has been issued, and DB2 has returned an SQLCODE of -805, which indicates that the program needs to be bound or rebound on that particular DB2 system.

System action:
A return code of 12 is set, and processing stops.

User response:
Bind or rebind the named module, and resubmit the job.

ADB9010E A plan access error occurred for program *program_name* because you are not authorized to run the plan.

Explanation:
The identified program did not run successfully because the program attempted to issue an SQL request, and DB2 issued an SQLCODE of -922.

System action:
A return code of 12 is set, and processing stops.

User response:
Correct the authorization, and resubmit the job.

ADB9011E An unexpected sqlcode was found in *error_function*.

Explanation:
This message is issued when the environment in which the program is running is not correct or a possible user error exists.

System action:
A return code of 12 is set, and processing stops.

User response:
Obtain a dump, and contact IBM Software Support.

ADB9012E The DD statement *ddname* is missing or is incorrect.

Explanation:
The JCL for the job is missing the identified DD statement or the DD statement is incorrect.

System action:
A return code of 12 is set, and processing stops.

User response:
Supply the missing DD statement, and resubmit the job.

ADB9013E The specified scope *scope_qualifier.scope_name* was not found.

Explanation:
The request required the use of a version scope and could not be processed because the scope that was specified does not exist.

System action:
A return code of 8 is set, and processing stops.

User response:
Correct the scope qualifier, scope name, or both to identify a scope that exists, and resubmit the request.

ADB9014I The specified version *version_qualifier.version_name* was found in the database.

Explanation:
The request was processed because the specified version exists.

System action:
None.

User response:
None.

ADB9015E The specified version *version_qualifier.version_name* was not found in the database.

Explanation:
The request could not be processed because the specified version does not exist.

System action:
A return code of 8 is set, and processing stops.

User response:
Correct the version qualifier, the version name, or both to identify a version that exists, and resubmit the request.

ADB9016W **The specified version *version_qualifier.version_name* exists but its definition is empty or incomplete.**

Explanation:
The request might not have been processed accurately because the version is not defined correctly.

System action:
A return code of 4 is set, and processing continues.

User response:
Correct the version qualifier, the version name, or both and ensure that the version has version records.

ADB9017I ***program_name* - Export Version Files**

Explanation:
This report message identifies the DB2 Admin program that is being run to export version files.

System action:
None.

User response:
None.

ADB9019I **The number of version data records exported is *integer*.**

Explanation:
After the Db2 Admin Tool program to export version files completes, this report message is issued to indicate the number of version file records that were exported.

System action:
None.

User response:
None.

ADB9020I ***ADBCVIC or ADBCVIM* - Import Version Files**

Explanation:
This report message identifies the Db2 Admin Tool program that is being run to import version files.

System action:
Processing continues.

User response:
None.

ADB9021I **Version Import Complete. The import for version *ADBCVIC or ADBCVIM* completed successfully.**

Explanation:
This report message indicates that the DB2 Admin program to import version files ran successfully.

System action:
None.

User response:
None.

ADB9022E **An invalid version type was specified. The valid values are BASE and DELTA.**

Explanation:
The input job that Db2 Admin Tool generated contains an invalid value for the version type. The type of version must be either BASE or DELTA.

System action:
A return code of 12 is set, and processing stops.

User response:
Edit the input job to specify a valid version type, and resubmit the job. Report this internal error to IBM Software Support.

ADB9023E **An invalid action for a version file import was specified. The valid values are ADD and REPLACE.**

Explanation:
The input job that Db2 Admin Tool generated contains an invalid value for the action to take when importing a version file. The action for the import must be either ADD or REPLACE.

System action:
A return code of 12 is set, and processing stops.

User response:
Edit the input job to specify a valid action, and resubmit the job. Report this internal error to IBM Software Support.

ADB9024I **The scope ID for version scope *scope_qualifier.scope_name* is scope *scope_identifier*.**

Explanation:
This report message indicates that the version scope with the identified scope ID is being processed.

System action:
Processing continues.

User response:
None.

ADB9025I **The version will be replaced.**

Explanation:
A version is being created. A version with the specified name already exists and will be overwritten.

System action:
Processing continues.

User response:
None.

ADB9026E **The version already exists. It cannot be added.**

Explanation:
Db2 Admin Tool is trying to process a request to add a new version. The version cannot be created because a version with the specified qualifier and name already exists.

System action:
A return code of 8 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB9027E **The input file is empty. No records were found.**

Explanation:
Db2 Admin Tool is trying to process a request but the input file that describes the action that should be taken is empty.

System action:
A return code of 12 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB9028I **A version file was created from *DB2_subsystem_id* at *extract_time* by *extract_sqlid*.**

Explanation:
This report message provides information about the version file that is being processed. It displays the ID of the DB2 subsystem, the time the version file was extracted, and the ID of the user who ran the Db2 Admin Tool program to extract the version information.

System action:
None.

User response:
None.

ADB9029I **A version file was extracted from location *DB2_location* at *extract_time* by *extract_sqlid*.**

Explanation:
This report message provides information about the version file that is being processed. It displays the location of the DB2 subsystem, the time the version file was extracted, and the ID of the user who ran the Db2 Admin Tool program to extract the version information.

System action:
None.

User response:
None.

ADB9030E **The version file description is not available because the input file does not have a header record.**

Explanation:
Db2 Admin Tool is trying to process a version file but cannot because the input file does not have a header record.

System action:
A return code of 12 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB9031W **The input file is empty. No records were found.**

Explanation:
Db2 Admin Tool is trying to process a request but cannot because the input file is empty.

System action:
A return code of 4 is set, and processing continues.

User response:
Report this internal error to IBM Software Support.

ADB9302E **Change "*change_owner.change_name*" cannot be recovered because the following changes must be recovered first and either they do not have a recover change or they have a recover change that is not in the ANALYZED state. Owner.Name --**

Explanation:
An attempt is being made to recover a change that cannot be recovered because other changes must be recovered first and those changes either do not have a recover change or have a recover change that is not in the ANALYZED state. The accompanying messages provide a list of the changes that must be recovered first that either do not have a recover change or have a recover change that is not in the ANALYZED state.

System action:
Processing stops.

User response:
Create a new change to undo the changes for the specified changes.

ADB9032I The number of version data records imported is *integer*.

Explanation:
After the Db2 Admin Tool program to import version files completes, this report message is issued to indicate the number of version data records that were exported.

System action:
None.

User response:
None.

ADB9033I The SQLCA sqlcode is *sqlca.sqlcode*.

Explanation:
This message displays the SQLCODE that was returned.

System action:
None.

User response:
None.

ADB9034I ADBCVOB - Object Extraction Complete

Explanation:
This report message indicates that the Db2 Admin Tool program to extract objects completed successfully.

System action:
None.

User response:
None.

ADB9035I The number of objects that were found is *integer*.

Explanation:
After the Db2 Admin Tool program to extract objects completes, this report message is issued to indicate the number of objects that were processed.

System action:
None.

User response:
None.

ADB9036I ADBCVOB - Extract Version Objects.

Explanation:
This report message indicates that the DB2 Admin program that extracts the objects for a version has started.

System action:
Processing continues.

User response:
None.

ADB9037I ADBCVSX - Export Scope Objects

Explanation:
This report message indicates that the Db2 Admin Tool program that extracts version scopes has started.

System action:
Processing continues.

User response:
None.

ADB9038I ADBCVSX - Scope Export Complete

Explanation:
This report message indicates that the Db2 Admin Tool program that extracts version scopes has completed successfully.

System action:
None.

User response:
None.

ADB9039E A scope name was not specified

Explanation:
The Db2 Admin Tool program to extract a version scope could not run because the input to the program did not include the scope name.

System action:
A return code of 12 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB9040E A scope qualifier was not specified.

Explanation:
The Db2 Admin Tool program to extract a version scope could not run because the input to the program did not include the qualifier for the scope.

System action:
A return code of 12 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB9041I The scope *scope_qualifier.scope_name* was found in the database. Its scope ID is *scope_identifier*.

Explanation

The scope that was being processed was found, and it has the identified scope ID.

System action:
Processing continues.

User response:
None.

ADB9042I **The number of scope objects written is *integer*.**

Explanation:
After the Db2 Admin Tool program to process scope objects completes, this report message is issued to indicate the number of scope objects that were processed.

System action:
None.

User response:
None.

ADB9043I **Its scope ID is *scope_identifier*.**

Explanation:
A version scope with the identified scope identifier is being processed.

System action:
Processing continues.

User response:
None.

ADB9044I **The version will be added.**

Explanation:
The Db2 Admin Tool program that processes versions will add a version.

System action:
Processing continues.

User response:
None.

ADB9045I **It should be there.**

Explanation:
Db2 Admin Tool is attempting to replace an existing version file, but the version file being replaced does not exist.

System action:
Processing continues.

User response:
Verify that having the version file replaced is the action that you want. Ensure that the version qualifier and version name are correct if you want the version replaced. If you do want to replace an existing version file, change the action to ADD instead of REPLACE in the input parameters to the Db2 Admin Tool program.

ADB9046E **The specified version *version_identifier* was not found in the database.**

Explanation:
DB2 is attempting to replace an existing version file with a version file that is being imported, but the version file being replaced does not exist.

System action:
A return code of 12 is set, and processing stops.

User response:
Verify that having the version file replaced is the action that you want. Ensure that the version qualifier and version name are correct if you want the version replaced. If you do want to replace an existing version file, change the action to ADD instead of REPLACE in the input parameters to the Db2 Admin Tool program.

ADB9047I **The version ID is *version_identifier*.**

Explanation:
A version with the identified version ID is being processed.

System action:
Processing continues.

User response:
None.

ADB9048I **The specified version *version_qualifier.version_name* was not found in the database.**

Explanation:
The version that is being processed should replace an existing version, but that version does not exist.

System action:
A return code of 12 is set, and processing stops.

User response:
Correct the version qualifier, version name, or both to identify a valid version, and resubmit the request.

ADB9049I **Scope object records are being processed.**

Explanation:
The process to extract version scope object definitions has started.

System action:
Processing continues.

User response:
None.

ADB9050I **Version *version_qualifier.version_name* is being extracted.**

Explanation:
A version is needed to process the request, and the identified version is being extracted.

System action:
Processing continues.

User response:
None.

ADB9051E The version name, qualifier, or both for version ID *version_identifier* is null in the database.

Explanation:
Db2 Admin Tool is trying to replace a delta version file, but a delta version file is not found for the version identifier that is provided as input to the Db2 Admin Tool program.

System action:
A return code of 12 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB9052W No scope object records for scope ID *scope_identifier* were found in the database.

Explanation:
The version that was created might be incomplete because there were no objects defined for the scope that was specified for the version.

System action:
A return code of 4 is set, and processing continues.

User response:
Complete the definition of the scope by editing the scope and adding objects to it.

ADB9057W A version already exists with the specified version name.

Explanation:
Auto mode is in effect, so the base version will be created with a name like `AUTO:` and `CURTS`.

User response:
None.

ADB9059W The version level *version_level* *version_name* having a type of *version_type* is not supported.

Explanation:
An unknown version level for the specified version record was found in the database.

User response:
Verify that the unknown base version record is valid. Return code of 4 is set and processing continues.

ADB9060I The processing for the ignore or mask begins.

Explanation:

System action:
Processing continues.

User response:
None.

ADB9061E An error occurred while processing the ignore or mask request.

Explanation:
The Db2 Admin Tool program that processes ignores and masks has encountered an error.

System action:
A return code of 8 is set, and processing stops.

User response:
See the previously issued message, which provides details about the error.

ADB9062I The processing for the ignore or mask completed successfully.

Explanation:
This report message indicates that Db2 Admin Tool has completed the processing for the ignore or mask successfully.

System action:
None.

User response:
None.

ADB9063E The input parameter *input_keyword* for the ignore or mask request was not provided.

Explanation:
The ignore or mask could not be processed because the input information that the DB2 Admin program needs was not provided. This is an internal error.

System action:
A return code of 12 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB9064E *op_parameter_value* is not a valid value for the OP parameter.

Explanation:
The ignore or mask could not be processed because the input to the Db2 Admin Tool program that processes ignores and masks did not contain a valid value for the OP parameter. This is an internal error.

System action:
A return code of 12 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB9065E *type_parameter_value* is not a valid value for the Type parameter.

Explanation:
The ignore or mask could not be processed because the input to the Db2 Admin Tool program that

processes ignores and masks did not contain a valid value for the Type parameter. This is an internal error.

System action:
A return code of 12 is set, and processing stops.

User response:
Report this internal error to IBM Software Support.

ADB9067I **Ignore *ignore_owner.ignore_name* was inserted to database.**

Explanation:
The request to add an ignore in the Change Management database was successful.

System action:
None.

User response:
None.

ADB9068E **The definition of ignore *ignore_owner.ignore_name* is incomplete (no ignore lines exist).**

Explanation:
The ignore cannot be used because its definition is empty.

System action:
A return code of 12 is set, and processing stops.

User response:
Complete the definition of the ignore by editing the ignore and specifying ignore fields, and resubmit the request.

ADB9069E **Ignore *ignore_owner.ignore_name* does not exist.**

Explanation:
The request required the use of an ignore and could not be processed because the specified ignore does not exist.

System action:
A return code of 12 is set, and processing stops.

User response:
Ensure that the correct ignore owner, scope name, or both was specified. Or, create an ignore with the owner and name that was specified. Then, resubmit the request.

ADB9070I **Ignore *ignore_owner.ignore_name* was retrieved from database.**

Explanation:
The request was processed because the required ignore exists.

System action:
None.

User response:
None.

ADB9071I **Mask *mask_owner.mask_name* was inserted to database.**

Explanation:
The request to add a mask in the Change Management database was successful.

System action:
None.

User response:
None.

ADB9072E **Mask *mask_owner.mask_name* does not exist.**

Explanation:
The request required the use of a mask and could not be processed because the specified mask does not exist.

System action:
A return code of 12 is set, and processing stops.

User response:
Ensure that the correct mask owner or mask name was specified. Or, create a mask with the specified owner and name and resubmit the request.

ADB9073W **The definition of mask *mask_owner.mask_name* is incomplete (no mask lines exist).**

Explanation:
The mask cannot be used because its definition is empty.

System action:
A return code of 4 is set, processing continues, and no system action taken.

User response:
If you do not intend to use the empty mask, complete the definition of the mask by editing the mask and specifying mask lines. Then, resubmit the request.

ADB9074IE **Mask *mask_owner.mask_name* was retrieved from the database.**

Explanation:
The request was processed because the required mask exists.

System action:
None.

User response:
None.

ADB9075I **The processing for an ignore or mask is ending.**

Explanation:
This report message indicates that Db2 Admin Tool has finished processing an ignore or a mask.

System action:
None.

User response:
None.

ADB9076E **The DD statement for *dd_name* is missing.**

Explanation:
The JCL for the job is missing the identified DD statement.

System action:
A return code of 12 is set, and processing stops.

User response:
Supply the missing DD statement, and resubmit the job.

ADB9078E **The specified base version *owner*, *name* has an unsupported version level: *version_level*.**

Explanation:
The specified base version cannot be used because it contains an earlier version level than the currently supported version. The version level of the base version is located in the CM ADBCVERSION table, TYPE='B'.

System action:
The error message is displayed. Return to the previous panel to restart the process.

User response:
Create the CM version again using the current release. Admin tool will re-create a new version level.

ADB9110I **The status of the following changes will be set to DEFINED:**

Explanation:
When a recover change is being run, any pending changes to the objects within the recover change are set to DEFINED status. The original change of the recover change is also set to DEFINED status. The original change supersedes any pending changes for the objects within the original change. The pending changes that were superseded are set to DEFINED status. This message introduces the list of the changes that are set to DEFINED status. Message ADB9113 is issued after this message to list each change that is set to DEFINED status.

System action:
Processing continues.

User response:
Review the list of changes that is displayed after this message to understand which changes are set to DEFINED status when the change is recovered.

ADB9111I **Owner.Name**

Explanation:
Messages ADB9110, ADB9111, ADB9112, and ADB9113 are issued in conjunction with each other. This message provides a heading to identify the owner and the name of the changes that are listed by message ADB9113.

System action:
Processing continues.

User response:
See message ADB9110.

ADB9112I **----——--**

Explanation:
Messages ADB9110, ADB9111, ADB9112, and ADB9113 are issued in conjunction with each other. This message provides a heading for message ADB9113.

System action:
Processing continues.

User response:
See message ADB9110.

ADB9113I ***change_owner.change_name***

Explanation:
Messages ADB9110, ADB9111, ADB9112, and ADB9113 are issued in conjunction with each other. This message lists the owner and name of each change that is set to DEFINED status when you recover the change.

System action:
Processing continues.

User response:
See message ADB9110.

ADB9300E **Change *change_owner.change_name* cannot be recovered until the following changes are recovered in the order that they are specified. The list contains those changes that completed after the change to recover completed and have not been recovered. They modify the same or related objects as those in the change to recover and, hence, the recover change itself. Rcvr Order Owner.Name ----- -- ---**

Explanation:
An attempt is being made to recover a change that cannot be recovered because other changes must be recovered first. The accompanying messages provide a list of the changes that must be recovered first.

System action:
Processing stops.

User response:
Recover the list of changes in the order that is specified.

ADB9304E **This change cannot be recovered because it does not have a recover change or its recover change is not in the ANALYZED state.**

Explanation:
An attempt is being made to recover a change that cannot be recovered because it does not have a recover change or its recover change is not in ANALYZED status.

System action:
Processing stops.

User response:
Ensure that each change currently being recovered that is, the change is in RUNNING status) completes. Otherwise, create a new change to undo the changes made by this change.

ADB9305I **The following pending changes will be set to DEFINED status. These changes modify the same or related objects as those in the change to recover and, hence, the recover change itself. Owner.Name --**

Explanation:
A change is being recovered, and there are pending changes for the objects that are affected by the change to recover. The pending changes will be set to DEFINED status. The accompanying messages provide a list the changes that will be set to DEFINED status.

System action:
Processing continues.

User response:
None.

ADB9306I **This change can be recovered. No other changes that modify the same or related objects completed after the change completed, and there are no pending changes that modify the same or related objects.**

Explanation:
A change is being recovered, and this informational message indicates that there are no other changes that need to be recovered first and that there are no pending changes for the affected objects.

System action:
Processing continues.

User response:
None.

ADB9307E **This change cannot be recovered because the WSL and JCL files for the recover change do not exist.**

Explanation:
An attempt was made to recover a change, and the WSL and JCL files that are required to recover the change do not exist. The change cannot be recovered.

System action:
Processing stops.

User response:
Create a new change to undo the changes made by this change.

ADB9308E **The JCL file for the recover change does not exist. An error occurred while a temporary JCL file for the recover WSL was being created.**

Explanation:
An attempt was made to recover a change, and the JCL file for the recover job that is required to recover the change does not exist.

System action:
Processing stops.

User response:
Create a new change to undo the changes made by this change.

ADB9351E **An error occurred when the change status was updated. Neither the old or new change status values match the current change status: *current_change_status*.**

Explanation:
The request to update the change status was invalid.

System action:
Processing stops.

User response:
If you submitted a run job, ensure that you analyze the change before running it. If you submitted an analyze job, ensure that the change is in DEFINED or ANALYZED status before submitting the analyze job.

ADB9352E **The specified change *change_ID* does not exist.**

Explanation:
A request was made to update the change status for a change ID that does not exist.

System action:
No system action is taken.

User response:
Try generating a new run job or re-analyze the change.

ADB9353E SQL error *SQL_error_code* occurred while the Change Management database was being accessed.

Explanation:
An unexpected SQL error occurred while accessing DB2.

System action:
None.

User response:
Fix the problem and try again.

ADB9355I The supersede information was cleared for this change.

Explanation:
This change is no longer superseded by another change. Any prerequisite changes still exist.

System action:
Processing continues.

User response

No action is required.

Related concepts

“Change Management terminology” on page 639
Before you use Change Management (CM), you should understand the terms that CM uses.

ADB9356E The change was superseded by change *ID*(*owner.name*) on *timestamp* and needs to be re-analyzed.

Explanation

The change cannot be run, because it is not in ANALYZED status. The status of the change was set to DEFINED by the identified change that supersedes it.

ID
The ID of the superseding change.

owner
The owner of the superseding change.

name
The name of the superseding change.

timestamp
The time at which the superseding change was created.

System action:
Processing stops.

User response

Analyze the change again.

Any prerequisite changes still exist and must be run before you can run this change.

Related tasks

“Analyzing a change” on page 652
After a change is registered, you must analyze it before you can run it. During this analyze step, Db2 Admin Tool analyzes how the change modifies existing objects and creates a work statement list (WSL) that can be used to run the change.

Related reference

“Types of changes and change status” on page 644
To facilitate change management, Db2 Admin Tool categorizes changes into several types and assigns a status to each change as it moves through the change management process.

ADB9400I The change was registered successfully. Changeid: *Change_ID*

Explanation:
The specified change was successfully registered.

System action:
No action is required.

User response:
Processing continues.

ADB9401E Registration has failed. Error in input parameters: Change Owner: *Change_Owner* Change Name: *Change_Name* Change Type: *Change_Type*

Explanation:
There was an error in one of the input parameters and the registration has failed.

System action:
Processing stops.

User response:
Correct the parameters and try again.

ADB9403E Registration has failed. Error in input parameters: Start Mode: *Start_Mode* Register Mode: *Register_Mode* Input Type: *Input_Type* Input Name: *Input_Name*

Explanation:
An error in one or more of the input parameters has caused the registration to fail.

System action:
Processing stops.

User response:
Correct the parameters and try again.

ADB9405E **Error registering the change. Another change already exists with: Change Owner: *Change_Owner* Change Name: *Change_Name* Change Type: *Change_Type***

Explanation:
The change cannot be registered because the change owner, name or type already exists.

System action:
Processing stops.

User response:
Modify the change owner and/or name and try again.

ADB9406E **Change does not exist. Change Owner: *Change_Owner* Change Name: *Change_Name* Change Type: *Change_Type***

Explanation:
The change must exist for including into an existing change.

System action:
Processing stops.

User response:
Ensure that the change already exists.

ADB9407E **ChangeID for the original change must be provided to recover. ChangeID: *Change_ID***

Explanation:

System action:
Processing stops.

User response:
Provide the changeid for the orginal change and try again.

ADB9409E **Registration could not be completed. Reason Code: *Reason_Code* Reason: *Reason* Change ID: *Change_ID***

Explanation:
The registration could not be completed for the specified reason.

System action:
Processing stops.

User response:
Correct the error and try again.

ADB9410E **The restart failed. A change ID is required to restart a change.**

Explanation:
You must specify the change ID of the change to restart.

System action:
Processing stops.

User response:
Specify the change ID of the change to restart.

ADB9411E **The change is not in restartable status. Change Status: *Change_Status***

Explanation:
Changes in INITIAL, DEFINED or ANALYZED status are eligible for restart.

System action:
Processing stops.

User response:
Ensure that the change is in restartable status.

ADB9412E **Too few parameters were specified to associate a target. Target Name: *Target_Name***

Explanation:
You must specify the correct number of parameters for the specified target.

System action:
Processing stops.

User response:
Specify the missing parameters and try again.

ADB9413E **The specified target is already associated with the MT Change. Target Name: *Target_Name* Target Change Owner: *Target_Change_Owner* Target Change Name: *Target_Change_Name* Target Change Status: *Target_Change_Status***

Explanation:
The specified target is already associated with the multi-target change.

System action:
Processing stops.

User response:
Specify a different target profile and try again.

ADB9414E **The target profile was not found. Target Profile Name: *Target_Profile_Name***

Explanation:
The specified target profile name was not found

System action:
Processing stops.

User response:
Specify an existing target profile and try again.

ADB9418E A multi-target change is already registered that uses either the same mask or no mask was specified. Details of the existing change: Change ID *change_ID*, Change Owner: *change_owner*, Change Name: *change_name*, Change Status: *change_status*.

Explanation:
You cannot use the same mask multiple times because it might result in redundant changes to objects.

System action:
Processing continues with the next change.

User response:
Specify a different mask and try the operation again.

ADB9419I An existing target change was restarted. Change ID: *change_ID*.

Explanation:
A request to register a multi-target change has been received; however, a change with the same mask already exists in INITIAL status. An attempt was made to restart the existing change rather than registering it as a duplicate change. The success or failure of restarting the change is reported.

System action:
Processing continues.

User response:
None required.

ADB9421E Cannot replace a change with Change Type: *Change_Type*.

Explanation:
A request to replace a change was received but cannot be processed. Only changes with the change type 'CHANGE' can be replaced.

System action:
Processing stops.

User response:
Modify the change owner or change name to select another existing change that has the change type 'CHANGE', or to create a new change and then try again.

ADB9422E Cannot replace a change with status: *Change_Status*.

Explanation:
A request to replace a change was received but cannot be processed because of an existing change. The existing change must have a change status of initial, defined, analyzed, or canceled in order to be replaced.

System action:
Processing stops.

User response:
Modify the change owner or change name to select another existing change that has the change type 'CHANGE', or to create a new change and then try again.

ADB9424E Registration failed to replace the change. Change ID . . . . : *change_ID*, Change Owner . . . . : *change_owner*, Change Name . . . . : *change_name*

Explanation:
Error occurred replacing a change. Review other messages in the report to ascertain the failure.

System action:
Processing stops.

User response:
Look for other messages that can help identify the reason that the replace change request failed. Correct the error and try again.

ADB9426E Check the Work Load Manager (WLM) environment started task *wlm_environment_name* for additional messages and check the WLM settings.

Explanation:
The call to the ADBCRSP procedure (the multiple target change stored procedure) failed.

System action:
Processing terminates abnormally.

User response:
Check the task started in the Workload Manager (WLM) environment *wlm_environment_name* for additional messages. Also, check with the User's Guide to confirm that the WLM settings are correct.

ADB9427E Cannot connect to the DB2 subsystem. Restore the connection and rerun the job.

Explanation:
The target change cannot be registered because there is a disconnection between the multi-target change program and the DB2 subsystem.

System action:
Processing stops.

User response:
Restore the connection and rerun the job.

ADB9428E A multi-target change cannot be registered because the connection to DB2 system no longer exists.

Explanation:
A failure has disconnected the application, so the application can not execute any SQL statement.

System action:
Processing stops.

User response:
Restore the connection and rerun the job.

ADB9722E **The value of the following parameter is invalid. Parameter name: *parameter_name* Value specified: *parameter_value***

Explanation

The specified parameter value is not valid. Often, this parameter is a Change Management (CM) batch parameter.

parameter_name
The name of the parameter.

parameter_value
The parameter value.

System action:
Processing stops.

User response

Fix the parameter value and try again.

If the parameter is a CM batch parameter, ensure that any documented requirements for the specified value are met. For example, some CM batch parameters require that another CM batch parameter be specified with a certain value.

Related reference
“CM batch parameter definitions” on page 664
You can use Change Management (CM) batch interface parameters to control Change Management (CM) actions and settings.

ADB9723E ***option_1* and *option_2* are mutually exclusive options**

Explanation:
The two options listed (*option_1* and *option_2) cannot be specified together.*

System action:
Processing stops.

User response:
Correct the specification.

ADB9735E **You requested that the DDL be generated from a base version, but the version does not exist**

Explanation:
If the type is USER, the owner and name values are the base version owner and name that you specified. Otherwise, the type indicates the type of base version you requested along with the change owner and name values that you specified.

System action:
Processing ends.

User response:
Ensure that the specified base version type exists for the specified change. If the DDL from a user-specified base version was requested, ensure that the version exists.

ADB9736E **You requested that the DDL be generated from a base version, but the version requested is not a base version.**

Explanation:
If the type is USER, the owner and name values are the base version owner and name that you specified. Otherwise, the type indicates the type of base version you requested, along with the change owner and name values that you specified.

System action:
Processing ends.

User response:
If the DDL from a user-specified base version was requested, ensure that the version owner and name you specified matches an existing base version and not a delta version. If you did not request the DDL from a user-specified base version, you should report this to IBM.

ADB9751E **The multi-target change cannot be analyzed because at least one action parameter other than action_analyze_change is set to Y.**

Explanation:
Analyzing a multi-target change will pick up all the corresponding target changes on the local DB2 subsystem for analysis. However, you cannot combine this action with other actions like run or recover.

System action:
Processing stops.

User response:
Fix the parameter value and try again.

ADB9752E **The change management batch parameter option *<option>* is not supported for the multi-target change analyze or run process.**

System action:
Processing stops.

User response:

Remove the parameter value and try again.

ADB9755W **Validate_ddl is forced to NO because validate DDL is strictly limited to validating DDL. Input file provided is a changes file.**

System action:
Only DDL is validated when validate_ddl = 'y'.

System action:
Processing continues.

User response:
If validation is required, specify a DDL file as the import file.

ADB9756I **Prerequisite changes may be present but validate DDL does not take them into consideration.**

Explanation:
Prerequisite changes are not seen by the DDL validation for import.

System action:
Processing continues.

User response:
No action is required.

ADB9757E **SOURCE_TYPE and/or TARGET_TYPE is set to 'USER'. When parsing the input for quick scopes, an error was encountered. The quick scope parameter, TYPE, is not present in the input. Note that DDL is not valid in the context of a quick scope.**

Explanation:
An incorrect scope was specified for SOURCE_TYPE=USER or TARGET_TYPE=USER.

System action:
Processing stops.

User response:
Fix the quick scope input and try again.

ADB9908I **ADB9908I Processing change: Owner . . : *owner_name*, Name : . . .*name***

Explanation:
Data for the identified change is being moved from the local backup tables into the identified InfoSphere Optim Configuration Manager repository database.

System action:
No system action is taken.

User response:
No action to take.

ADB9909I **Statement information: Approximate run timestamp . . :*timestamp*, Statement type *statement_type* Object type *object_type,* , Object qualifier *object_qualifier*, object name *object_name***

Explanation:
Data for the identified statement is being moved from the local backup tables into the identified InfoSphere Optim Configuration Manager repository database.

System action:
No system action is taken.

User response:
No action to take.

ADB9910E **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database. A severe error occurred.**

Explanation:
The InfoSphere Optim Configuration Manager repository database is not available. Look for other SQL error messages for the details of the error.

System action:
Processing stops.

User response:
Correct the error and try again.

ADB9911E **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database. The action on error setting is *action_on_error*. A ROLLBACK will be done and processing will stop.**

Explanation:
The InfoSphere Optim Configuration Manager repository database is not available. Look for other SQL error messages for the details of the error.

System action:
Processing stops.

User response:
Correct the error and try again.

ADB9912W **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database. The action on error setting *action_on_error*. The data was stored into backup tables on the local system. Processing continues.**

Explanation:
The InfoSphere Optim Configuration Manager repository database is not available. Look for other SQL error messages for the details of the error. The data was instead stored in backup tables on the local system.

System action:
Processing continues.

User response:
When the InfoSphere Optim Configuration Manager repository database is available, the DB2 Admin ADBLIM program can be run to move data from the backup tables on the local system to the InfoSphere Optim Configuration Manager repository database.

ADB9913E **The change information could not be stored into the backup tables on the local system. A severe error occurred.**

Explanation:
A severe error occurred while attempting to write to the backup tables on the local system. Look for other SQL error messages for the details of the error.

System action:
Processing stops.

User response:
Correct the error and try again.

ADB9914E **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database, or in the backup tables on the local system. The action on error setting is *action_on_error*. Processing stops.**

Explanation:
The InfoSphere Optim Configuration Manager repository database is not available, and the backup tables on the local system are not available. Look for other SQL error messages for the details of the error.

System action:
Processing stops.

User response:
When the InfoSphere Optim Configuration Manager repository database is available, the DB2 Admin ADBLIM program can be run to move data from the backup tables on the local system to the InfoSphere Optim Configuration Manager repository database.

ADB9915E **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database, or in the backup tables on the local system. The action on error setting is OVERRIDE. Processing stops. You can specify to override the error and continue processing the change. If the OCM repository database and the backup tables on the local system are not available, DB2 Admin will continue processing the change but information about the change will not be recorded. To override the error in batch: - When using CM batch, specify the OVR_CONFIGDB_ERROR = 'Y' parameter in the PARMS DD file. When not using CM batch, specify the OVR_CONFIGDB_ERROR = 'Y' parameter in the ADBTEPIN DD file. To override the error online, specify YES to the override option.**

Explanation:
The InfoSphere Optim Configuration Manager repository database is not available, and the backup tables on the local system are not available. Look for other SQL error messages for the details of the error.

System action:
Processing stops.

User response:
When the InfoSphere Optim Configuration Manager repository database is available, the DB2 Admin ADBLIM program can be run to move data from the backup tables on the local system to the InfoSphere Optim Configuration Manager repository database.

ADB9916W **The change information could not be stored into the InfoSphere Optim Configuration Manager repository database, or in the backup tables on the local system. The action on error setting is *action_on_error*. The OVR_CONFIGDB_ERROR parameter was set to 'YES', so the information about the changes made will not be stored in the InfoSphere Optim Configuration Manager repository database, or the local backup tables.**

Explanation:
The InfoSphere Optim Configuration Manager repository database is not available, and the backup tables on the local system are not available. Look for other SQL error messages for the details of the error.

System action:
Processing continues.

User response:
Once the InfoSphere Optim Configuration Manager repository database is available, the DB2 Admin ADBLIM program can be run to move data from the backup tables on the local system to the InfoSphere Optim Configuration Manager repository database.

ADB9918W For SHRLEVEL CHANGE processing, RECLUSTER NO is always enforced by the REORG TABLESPACE utility.

System action:
Processing continues.

User response:
No action is required.

ADB9929E Change *change_owner.change_name* was not deleted. *Reason*

Explanation

The specified change was not deleted for one of the following reasons:

- You do not have the authority to run delete change command.
- Delete change command is not enabled.
- Change does not satisfy the delete criteria.
- User does not have the privilege to delete change according to the definition of the delete change view (ADBCHGV1).
- The delete change function was not successful due to an unexpected error condition.

System action:
Processing stops.

User response:
Check the reason, fix the problem, and run again.

ADB9962I The change that was specified is empty, so you not need to generate base versions. The following generate base version variables have been reset to NO: *parameters*

Explanation

Because the change is empty, base versions are not needed and the listed parameters are set to NO.

parameters
The Change Management (CM) batch parameter names.

System action:
Processing continues.

User response

No action is required.

Related reference

“CM batch parameter definitions” on page 664
You can use Change Management (CM) batch interface parameters to control Change Management (CM) actions and settings.

ADB9963I This report displays totals for all of the *items_counted*.

Explanation:
During the CM Analyze phase, the totals report includes counts for various object types. The counts are for the number of changed objects. During the CM Run phase, the totals report includes counts for various object types. The counts are for the number of SQL statements issued. For more information about the Totals Report, see Examples: Invoking the Change Management batch interface for various actions.

System action:
Processing continues.

User response:
None.

ADB9966I Compare CHG . . . . . : *data_set_name*

System action:
Processing continues.

User response:
None.

ADB9967I Compare IFF PDS . . . : *data_set_name*

System action:
Processing continues.

User response:
None.

ADB9968E The *action_parameter_name* action parameter is enabled. This action parameter can be enabled only when the following action parameter is enabled.

System action:
Processing stops.

User response:
Fix the value of the parameter, and run the process again.

ADB9969E The *action_parameter_name* action parameter is enabled. This action parameter is in conflict with at least one other parameter that is enabled.

System action:
Processing stops.

User response:
Fix the value of the parameter, and run the process again.

ADB9972E **Convert to Read WSL Dataset not found: dataset_name.**

Explanation:
The WSL to be converted to readable format was not found.

System action:
Processing stops.

User response:
Ensure that the WSL is specified correctly and that a closing quotation mark is included.

ADB9973E **The change_owner and change_name are required.**

Explanation:
The PDS_FOR_WSL parameter was not specified. The default response is to use the WSL that is specified in the change, but values for change_owner and change_name were not specified either.

System action:
Processing stops.

User response:
Specify the name of PDS_FOR_WSL parameter or specify both the change_owner and change_name parameters.

ADB9974E **The WSL name was not specified in the change.**

Explanation:
The change definition did not include the name of the WSL data set.

System action:
Processing stops.

User response:
Specify the name of the WSL in the change definition.

ADB9975E **Error retrieving the change information.**

Explanation:
The named change could not be retrieved from the ADBCHG table.

System action:
Processing stops.

User response:
Verify the name of the change.

ADB9976E **PDS_for_wsl_conv is required when worklist_name_conv is specified.**

Explanation

When the work list name is specified, the PDS name must be specified too.

System action:
Processing stops.

User response:
Specify the PDS_for_wsl_conv parameter and rerun the job.

ADB9977E **A data set error occurred: *error_text***

Explanation

A data set allocation error occurred. See the text for information.

System action:
Processing stops.

User response:
Resolve the problem described in the message, and submit the job again.

ADB9980E **Parameter action_generate_jcl_from_wsl = 'Y' is only valid under certain conditions: compare = 'Y' and action_generate_wsl = 'Y' pds_for_wsl is optional in this case action_compare = 'N' and action_generate_wsl = 'N' pds_for_wsl is required in this case worklist_name is always required and also used for the JCL.**

Explanation

The CM batch parameter action_generate_jcl_from_wsl can be set to Y only in the following two situations:

- When the following CM batch parameter values are set:

  ```
  compare = 'Y'
  action_generate_wsl = 'Y'
  ```

  In this case, the pds_for_wsl parameter is optional.
- When the following CM batch parameter values are set:

  ```
  action_compare = 'N'
  action_generate_wsl = 'N'
  ```

  In this case, the pds_for_wsl parameter is required.

In both situations, the worklist_name parameter is also required.

System action:

Processing stops.

User response:
Correct the parameter values to be a valid combination.

Related reference

“CM batch parameter definitions” on page 664
You can use Change Management (CM) batch interface parameters to control Change Management (CM) actions and settings.

ADB9981E Table ADBCHGAT could not be updated for a target change. Details of the target change: Change owner . . . : *owner* Change name . . . : *name*

Explanation

This error can occur when trigger ADB_MT_CHANGE_UPDATE attempts to call stored procedure ADBCRSU and fails.

owner
The change owner.

name
The change name.

System action:
Processing stops.

User response:
Confirm that ADB_MT_CHANGE_UPDATE and ADBCRSU are defined and started correctly.

ADB9982E generate_job_class is set to Y and one of the job_card_line parameters also contains the CLASS parameter

Explanation:
Having generate_job_class = Y while one of the job_card_line parameters also contains the CLASS parameter might generate a job that results in a JCL error.

System action:
Processing stops.

User response:
Set generate_job_class to N or remove CLASS from the job_card_line parameters.

Related information

“GENERATE_JOB_CLASS” on page 700

“JOB_CARD_LINE_1” on page 701

ADBA015E The list cannot contain the PARTLEVEL keyword.

Explanation:
For lists that are used by the MODIFY STATISTICS utility, the LISTDEF utility statement cannot contain the PARTLEVEL option. This specification is restricted by Db2.

System action:
Processing stops.

User response:
Correct the LISTDEF statement so that it does not include the PARTLEVEL option.

Related concepts

“LISTDEF and TEMPLATE” on page 416
LISTDEF and TEMPLATE are Db2 utilities that provide facilities for other utilities. You can use LISTDEF to define reusable lists of objects for other utilities to process. You can use TEMPLATE to define templates for data sets that are allocated by other utilities. LISTDEF and TEMPLATE are often used together.

Related information

Syntax and options of the MODIFY STATISTICS control statement (Db2 12 for z/OS)

ADBA016W For SHRLEVEL CHANGE processing, RECLUSTER NO is always enforced.

Explanation:
This warning message indicates that RECLUSTER NO is always enforced for SHRLEVEL CHANGE processing.

System action:
Processing continues.

User response:
No action is required.

ADBC099E There is a WSL mismatch. The WSLs did not compare equally.

Explanation:
The run-time WSL and the analyze-time WSL are different.

System action:
Processing stops.

User response:
Examine the environment to determine whether the change needs to be re-analyzed.

ADBC007E Invalid field name in the IGNORES file record.

Explanation:
The IGNORES file contains invalid ignore field specifications which can not be processed.

System action:
Processing stops.

User response:
Review the ignore field specifications in the IGNORES file and make sure all the fields specified are listed as

supported catalog table ignore fields or redefine the ignores by specifying the ignore fields on the Specify Ignore Fields panel.

ADBC016E **The object *object_owner*.*object_name* exists.**

Explanation:
An object *object_owner*.*object_name* already exist. Specify a new owner and name.

System action:
Processing stops.

User response:
Specify an owner and name so that the combination of owner and name is unique from objects that already exist.

ADBC027E **Target profile not found**

Explanation:
The specified target profile, **`target_profile`** was not found.

System action:
Processing ends.

User response:
Specify a valid target profile and try the operation again.

ADBC030E **Register Failed.**

Explanation:
The reason code and reason for failure are displayed as part of the long message. If the failure occurred during a call to a register interface, the return code from the interface is displayed as the reason code.

System action:
The process of registering a change terminated.

User response:
If the reason information does not help to resolve the issue, contact IBM® support to report the message.

ADBC060E **Pending changes exist that have an incompatible record layout.**

Explanation:
One or more pending changes exist that have an internal record layout that is incompatible with the current version of the product.

System action:
Processing is halted to prevent use of the incompatible records.

User response:
You can use the RST line command from the change management dialog (panel ADB2C11) to reset each of the incompatible record layouts.

ADBC066W **No target changes to process.**

Explanation:
An attempt was made to export multi-target information into a dataset on the target system in an effort to communicate target updates to the central system.

System action:
Processing stops. No information is written to the TGTINFO file.

User response:
Ensure that a list of multi-target changes is provided.

ADBC068E **The specified base version *owner*, *name* has an unsupported version level: *version_level*.**

Explanation:
The version level is not supported.

System action:
Processing stops.

User response:
Specify the appropriate version level and try again.

ADBC081 **The JCL data set or data set member does not exist.**

Explanation:
This message is issued when the ER line command was issued to edit the run job or promote job or the EA line command was issued to edit the analyze job for a change, and the JCL data set or data set member does not exist.

System action:
Processing stops

User response

If the ER line command was issued, the action to take depends on the type of change and its status:

- For a change type of CHANGE:
  - If the status is ANALYZED, issue RN line command to rebuild the run job.
  - If the status is RUNNING, use the job that is stored in SDSF.
  - If the status is not ANALYZED or RUNNING, the run job is no longer needed and no action is needed.
- For a change type of COMPARE:
  - If the status is DEFINED, the promote job is no longer valid. Create the promote job again.
  - If the status is COMPLETE, the promote job is no longer needed because the job has already been executed successfully and no action is needed.

If the EA line command was issued, the action to take depends on the status of the change:

- If the status is ANALYZED, RUNNING, or COMPLETE, the analyze job that was used to analyze the change is not accessible. No action is needed.
- If the status is DEFINED, re-analyze the change.
- If the status is none of the above, get the change into DEFINED status and then re-analyze the change.

ADBC082 **The change cannot be run because there are prerequisite changes that must be run first.**

Explanation:
The RN line command was issued to build a run job, but the change has prerequisite changes that must be run first.

System action:
Processing stops.

User response:
Run the prerequisite changes before re-issuing the RN line command to build the run job for the change. You can issue the PQ line command on the **Changes** panel (ADB2C11) to get a list of the prerequisite changes.

ADBC083E **A RESTART parameter was not automatically added because the job card is missing. The job needs to be restarted.**

Explanation:
The ER line command was issued so that the job to run (or promote) a change in RUNNING status could be edited and then resubmitted. Db2 Admin Tool was unable to automatically add the RESTART parameter to have the job restarted at the identified step because the job card is missing.

System action:
Processing continues, and the JCL to run the job is displayed in edit mode.

User response:
Add a job card to the JCL that includes a RESTART parameter so that the job is restarted at the identified step. Then, submit the job.

ADBC084E **A RESTART parameter was not automatically added to restart the step that runs program ADBTEP2 because the step could not be found.**

Explanation:
The ER line command was issued so that the job to run (or promote) a change in RUNNING status could be edited and then resubmitted. Db2 Admin Tool was unable to automatically add the RESTART parameter to have the job restarted at the step that runs program ADBTEP2 because Db2 Admin Tool could not find the step that runs that program.

System action:
Processing continues, and the JCL to run the job is displayed in edit mode.

User response:
Ensure that the JCL is valid. Then, submit the job.

ADBC085E **The RESTART parameter was not automatically added to the job card because either the step that runs the program could not be found or the job card is missing.**

Explanation:
The ER line command was issued so that the job to run (or promote) a change in RUNNING status could be edited and then resubmitted. Db2 Admin Tool was unable to automatically add the RESTART parameter. Either the step that runs the identified program could not be found or the job card is missing.

System action:
Processing continues, and the JCL to run the job is displayed in edit mode.

User response:
Ensure that the JCL is valid. Then, submit the job.

ADBC100E **The *owner.name* change does not exist.**

Explanation:
An attempt was made to delete a change that does not exist.

System action:
Processing ends.

User response:
Refresh the panel to retrieve the current list of changes.

ADBC101E **You do not have the privilege to delete the *owner.name* change according to the definition of the delete change view (ADBCHGV1).**

Explanation:
The delete change view (ADBCHGV1) has been defined in a way that prevents you from deleting the change.

System action:
Processing ends.

User response:
Check with the system administrator who installed Db2 Admin Tool and enabled Change Management.

ADBC102E **The *owner.name* change cannot be deleted because the change does not satisfy the delete criteria.**

Explanation

The change cannot be deleted because the change does not meet the criteria for being dropped. To be dropped, a change must meet one of the following criteria:

- The status of the change is CANCELED
- The status of the change is FAILED and the type is FAST
- The type of the change is COMPARE

System action:
Processing ends.

User response:
Put the change into a status such that the criteria to delete a change is met, and then try the DEL line command to delete the change again.

ADBC103E **You do not have the privilege to run the delete change command.**

Explanation:
You have not been given the privilege to delete changes. This error usually means that an SQLCODE -922 was received while an attempt was made to run the ADBCDCH plan.

System action:
Processing ends.

User response:
Check with the system administrator who sets up the DB2 Admin plans and packages to request access to the ADBCDCH plan.

ADBC104E **The delete change command is not enabled.**

Explanation:
Db2 Admin Tool has not been configured to enable the delete change command. This error usually means that an SQLCODE -805 was received while an attempt was made to run the ADBCDCH package.

System action:
Processing ends.

User response:
Check with the system administrator who sets up the Db2 Admin Tool plans and packages to request the appropriate set up of the ADBCDCH package and plan.

ADBC144W **Dynamic SQL statements can be captured to the catalog, but Db2 does not use any captured access paths, because the value of the CACHEDYN_STABILIZATION subsystem parameter is CAPTURE. Consider setting its value to BOTH.**

Explanation:
Db2 cannot use the captured access paths.

System action:
Processing continues.

User response:
If you want Db2 to use captured access paths, consider changing the value of the CACHEDYN_STABILIZATION subsystem parameter to BOTH.

Related information

CACHEDYN_STABILIZATION subsystem parameter (Db2 12 for z/OS)

ADBC145W **Db2 uses the captured access paths for dynamic SQL statements, but statements cannot be captured for stabilization, because the value of the CACHEDYN_STABILIZATION subsystem parameter is LOAD. Consider setting its value to BOTH.**

Explanation:
Db2 cannot capture dynamic SQL statements for stabilization.

System action:
Processing continues.

User response:
If you want Db2 to capture dynamic SQL statements for stabilization, consider changing the value of the CACHEDYN_STABILIZATION subsystem parameter to BOTH.

Related information

CACHEDYN_STABILIZATION subsystem parameter (Db2 12 for z/OS)

ADBC154E **Incompatible data sets. A data set with multi-target change content and a data set without multi-target change content cannot be imported together.**

Explanation:
A data set with multi-target change content cannot be imported with other data set(s). A data set with multi-target change content must be imported alone.

System action:
Processing ends.

User response:
Perform the import using a single multi-target change content data set, and another import using all non-multi-target change data sets.

ADBC155E **Incompatible data sets. Multiple data sets with multi-target change content cannot be imported at the same time.**

Explanation:

A data set with multi-target change content cannot be imported at the same time as other data sets with multi-target change content.

System action:
Processing ends.

User response:
Import data sets with multi-target change content one at a time.

ADBC158E For a delta change, the data set must be either fixed length (F/FB) with an LRECL of 80 or variable length (V/VB) with an LRECL of 16384.

Explanation:
The specified data set for delta changes does not have the required format.

System action:
Processing stops.

User response:
Specify a data set with the required format.

ADBC195E TSNAME *database_name.tablespace_name* is not defined in DB2

Explanation

The listed table space does not exist in the Db2 catalog.

database_name.tablespace_name
The qualified table space name.

System action:
Processing stops.

User response:
Specify an existing table space.

ADBC301E The EDIT line command requires installation and enablement of the DB2 Table Editor.

Explanation:
The DB2 Table Editor is not installed and enabled.

System action:
Processing ends.

User response:
Check with the Tools Customizer administrator for the DB2 Admin for assistance.

ADBC302E The EDIT line command is not enabled.

Explanation:
The EDIT line command is not enabled.

System action:
Processing ends.

User response:
Check with the Tools Customizer administrator, and ensure that the **`Enable DB2 Table Editor`** parameter is set to YES.

ADBC303E The EDIT line command cannot locate the DB2 Table Editor library.

Explanation:
The EDIT line command cannot locate the DB2 Table Editor library.

System action:
Processing ends.

User response:
Check with the Tools Customizer administrator, and ensure that the DB2 Table Editor CLIST library has been specified.

ADBC313E Mask value error

Explanation:
The mask value in the To column cannot start with a comma.

System action:
Processing stops.

User response:
Remove the comma.

ADBC314E Mask value error

Explanation:
The mask value in the From column cannot end with a comma.

System action:
Processing stops.

User response:
Remove the comma.

ADBC318E The overwrite value specified for the *mask_type* should be *correct_value*. Mask type = *mask_type*. The current overwrite value is *current_mask_value*.

Explanation:
The overwrite value specified for the *mask_type* should be *correct_value*.

System action:
Processing stops.

User response:
Correct the definition of the mask and try again.

ADBC330E Admin Tool 12.1 does not support column names longer than 30

Explanation:
Db2 Admin Tool 12.1 cannot read the IFF file, because it includes a column name that is longer than 30 characters.

System action:
Processing stops.

User response:
You must use Db2 Admin Tool 13.1 to read this IFF file.

ADBE003E Invalid value. Specify a number with the unit DAY(S), MONTH(S), or YEAR(S).

Explanation

Examples of valid values are:

3 DAYS
9 MONTHS
1 YEAR

System action:
Processing stops.

User response:
Correct the value.

ADBE004E Invalid value. Specify a number with the unit MINUTE(S), HOUR(S), DAY(S), MONTH(S), or YEAR(S).

Explanation

Examples of valid values are:

1 MINUTE
5 MINUTES
7 HOURS
3 DAYS
9 MONTHS
1 YEAR

System action:
Processing stops.

User response:
Correct the value.

ADBG001E Verification of the data set failed. The input data set must be either fixed length (F/FB) with LRECL=80 or variable length (V/VB) with LRECL between 16000 and 16384.

Explanation:
The input dataset must be either fixed length with a record length of 80 bytes or variable length with record length between 16000 and 16384 bytes.

System action:
Processing stops.

User response:
Specify a valid record format (RECFM) and record length (LRECL) for the data set.

ADBG002E Verify failed -- Value for DSORG is not supported.

Explanation:
The data set is a type that cannot be processed.

System action:
Processing stops.

User response:
Specify a member name and try again.

ADBG004E No member name specified

Explanation:
A member name is required for the PDS or LIBRARY.

System action:
Processing ends.

User response:
Add a member name and try the operation again.

ADBG009E Invalid entry specified

Explanation:
Either an invalid directory block number was specified for the data set name type or an invalid data set name type was specified for the directory block number.

System action:
Processing ends.

User response:
Modify the directory blocks value and try the operation again.

ADBG010E Verification has failed

Explanation:
The value specified for LRECL, RECFM, or DSNTYPE does not match the value for the existing data set.

System action:
Processing ends

User response:
Modify the value for the parameter and try the operation again.

ADBG011E Data set does not exist

Explanation:
The specified data set or member does not exist.

System action:
Processing continues.

User response:
Ensure that the specified data set exists and try the operation again.

ADBG013E All columns have been deleted except for one or more hidden columns. A table cannot contain only hidden columns.

Explanation:

The table from which you are deleting columns contains hidden columns. In DB2, a table cannot contain only hidden columns.

System action:
None.

User response:
No action required.

ADBG014E The *attribute-type* is only valid with implicit table space creation. If you specify a *specification*, leave this field blank.

Explanation

Creation of a table in an explicitly created table space was requested, but a specified attribute value is valid only for a table in an implicitly created table space.

attribute-type
DSSIZE or PAGENUM.

specification
DSSIZE or PAGENUM, if the message is displayed in the Create Table Options panel.

DSSIZE, PAGENUM, or "table space", if this message is displayed in the Create Table panel.

System action:
Processing stops.

User response

In the Create Table panel:

- If you specify a table space name, do not specify DSSIZE or PAGENUM value.
- If you specify a DSSIZE or PAGENUM value, do not specify a table space name.

In the Create Table Options panel:

- If you specified a table space name in the Create Table panel, do not specify a DSSIZE or PAGENUM value.

ADBG015E A hash specification is not valid with the PAGENUM RELATIVE attribute.

Explanation:
In the Create Table panel, a value of RELATIVE was specified for PAGENUM, but the ORGANIZE BY HASH option was also specified for a column in the table. These options are incompatible.

System action:
Processing stops.

User response:
Do not specify ORGANIZE BY HASH for a column in a table for which you specify the PAGENUM RELATIVE option.

ADBG016E The PAGENUM attribute is only valid when PARTITION BY RANGE (PBR) is also specified.

Explanation:
In the Create Table Options panel, a value of RELATIVE was specified for the PAGENUM attribute, and a value of No was specified for the PBR field. A table with relative page numbering can be created only in a range-partitioned table space.

System action:
Processing stops.

User response:
Do not specify a PAGENUM value of RELATIVE when you specify a PBR value No.

ADBG017E A partition by range specification has been previously defined, and is not valid with a non-null PBG size value.

Explanation:
A table space was previously defined with a partition-by-range specification, so DB2 Administration Tool set the PBR value in the Create Table Options panel to Yes. However, the user subsequently set the PBR value to No or blank, and the PBG size value to a non-null value. These values are incompatible.

System action:
Processing continues.

User response:
Set the PBR value to Yes, and do not specify a value for PBG size.

ADBM001E Too many columns

Explanation:
The maximum number of ORDER BY columns that can be defined is 10.

User response:
Reduce the number of columns that have been selected, and try again.

ADBM002E Invalid column

Explanation:
The column with the name COLnnnn can not be used in an ORDER BY clause in DB2 Admin because the column is the result of an expression.

User response:
Remove the column from the list of columns that are designated to be saved in the ORDER BY clause.

ADBM003E ORDER command not valid

Explanation:

The ORDER command cannot be used on this panel because DB2 Admin requires that the rows be in a defined sequence.

User response:
Use valid commands to configure the current panel. Valid commands are listed on the panel.

ADBM005E Save failed

Explanation:
The ORDER BY clause was not saved. Examine the ISPF log data set.

User response:
See the error that was written in the ISPF log data set. Resolve the problem and retry.

ADBM006E ORDER BY error

Explanation:
The ORDER BY clause for the panel caused SQLCODE -208 and the column in error was removed from the SELECT statement. Remove the column from the ORDER BY clause by using the ORDER command.

User response:
Exit this panel and return to the previous panel to remove the column, and try again.

ADBM009E Promote failed

Explanation:
The promotion of the ORDER BY clause to the installation default data set failed. Examine the ISPF log data set.

User response:
See the error that was written in the ISPF log data set. Resolve the problem and retry.

ADBM024E The overwrite value that is specified for the SEGSIZE must be an integer that is a multiple of 4.

Explanation:
The mask contains a value for SEGSIZE that is not valid.

System action:
A return code of 1012 is set, and processing stops.

ADBM025E The overwrite value that is specified for COMPRESS must be YES or NO.

Explanation:
The mask contains a value for COMPRESS that is not valid.

System action:
A return code of 1012 is set, and processing stops.

User response:
Change the mask definition to specify a value for SEGSIZE that is a multiple of 4, and then resubmit the job.

ADBM026E The overwrite value for DSSIZE must be a numeric value that is followed by the character 'G'.

Explanation:
The use of masking was specified, and the value that is specified for DSSIZE is not valid.

System action:
A return code of 1012 is set, and processing stops.

User response:
Correct the definition of the mask. If a specific value is specified for DSSIZE, ensure that the value is an integer value that is followed by the character 'G', for example, 8G. If a REXX user exit is specified for DSSIZE, ensure that the REXX user exit is coded so that it returns an integer value followed with character 'G'. After the corrections are made, resubmit the job.

ADBM027E The overwrite value for *space_allocation_quantity_attribute* must be a numeric value.

Explanation:
The use of masking was specified, and the value that is specified for *space_allocation_quantity_attribute* (PRIQTY, TSPRIQTY, IXPRIQTY) is not valid.

System action:
A return code of 1012 is set, and processing stops.

User response:
Correct the definition of the mask. If a specific value is specified for *space_allocation_quantity_attribute*, ensure that the value is an integer value.
If a REXX user exit is specified for *space_allocation_quantity_attribute*, ensure that the REXX user exit is coded so that it returns an integer value. After the corrections are made, resubmit the job.

ADBM028E The overwrite value for *space_allocation_quantity_attribute* must be a numeric value.

Explanation:
The use of masking was specified, and the value that is specified for *space_allocation_quantity_attribute* (SECQTY, TSSECQTY, or IXSECQTY)) is not valid.

System action:
A return code of 1012 is set, and processing stops.

User response:
Correct the definition of the mask. If a specific value is specified for *space_allocation_quantity_attribute*, ensure that the value is an integer value.
If a REXX user exit is specified for *space_allocation_quantity_attribute*, ensure that the

REXX user exit is coded so that it returns an integer value. After the corrections are made, resubmit the job.

ADBM029E The overwrite value for DEFER must be YES or NO.

Explanation:
The use of masking was specified, and the value that is specified for DEFER is not valid.

System action:
A return code of 1012 is set, and processing stops.

User response:
Correct the definition of the mask. If a specific value is specified for DEFER, ensure that the value is YES or NO. If a REXX user exit is specified for DEFER, ensure that the REXX user exit is coded so that it returns the value YES or NO. After the corrections are made, resubmit the job.

ADBM030E The overwrite value for *define_attribute* must be YES or NO.

Explanation:
The use of masking was specified, and the value that is specified for *define_attribute* (DEFINE, TSDEFINE, or IXDEFINE) is not valid.

System action:
A return code of 1012 is set, and processing stops.

User response:
Correct the definition of the mask. If a specific value is specified for *define_attribute*, ensure that the value is YES or NO. If a REXX user exit is specified for *define_attribute*, ensure that the REXX user exit is coded so that it returns the value YES or NO. After the corrections are made, resubmit the job.

ADBM209E The column is a pending drop column so it cannot be dropped.

Explanation:
The DROP line command was entered on the ADB21TC panel but the column is already marked to be dropped and the table space is in advisory REORG-pending status.

System action:
Processing stops.

User response:
A column marked as pending drop cannot be dropped. Select a different valid column.

ADBM703E The selected dialog name does not exist.

Explanation:
The selected dialog name might have been renamed or deleted by another user.

System action:
Processing stops.

User response:
Enter REFRESH on the command line, and then select a dialog name that is available.

ADBM706E The *&zcmd* command cannot be used with the line command that you specified. Remove the *&zcmd* command and then proceed.

Explanation:
The command cannot be used with the line command.

System action:
Processing stops.

User response:
Remove the command and press Enter. The product will continue to execute the line commands one by one.

ADBM708E A template syntax error occurred while building the apply job or work statement list.

Explanation:
Db2 Admin Tool detected a syntax error for the user-defined template while building the apply job or work statement list.

System action:
No system action is taken.

User response:
Modify the definition for the user-defined template.

ADBP01AW Target sequence name *sequence_name* will be substituted for the alias name in the ALTER statement. Press ENTER to continue.

Explanation:
This warning message indicates that the target sequence name will be used instead of the sequence alias name in the ALTER statement that is built and executed. The effect is the same as using the alias name; both act upon the base object.

System action:
The warning message is displayed on the panel.

User response:
Press Enter to continue processing.

ADBP01CE The referenced table cannot be *table_type*.

Explanation

The table type cannot be the following types of tables:

- An auxiliary table.

- A table that was implicitly created for an XML column.

System action:
The panel is redisplayed with the error message.

User response:
Choose a valid table type.

ADBP010E **A LOB table space must be defined before defining the auxiliary table.**

Explanation:
You tried to define an auxiliary table for the new LOB column. However, the associated LOB table space is not yet defined. The LOB table space must be defined before the auxiliary table is defined.

System action:
Processing stops.

User response:
Use the CS line command to define a LOB table space. Then retry the CA line command.

ADBP011E **An auxiliary table for the LOB column must be defined before defining the index on the table.**

Explanation:
You tried to define an index on an auxiliary table that is not yet defined. The auxiliary table for the new LOB column must be defined before the index is defined.

System action:
Processing stops.

User response:
Use the CA line command to define an auxiliary table. Then retry the CX line command.

ADBP012E **All explicit LOB objects must be defined for each new LOB column. One or more of the following definitions are missing: A LOB table space, an auxiliary table, or an index on the auxiliary table.**

Explanation

If you choose to explicitly define objects for a new LOB column, you must define all of the following objects for the new LOB column:

- the LOB table space
- the auxiliary table
- the auxiliary index

At least one of those objects is not defined.

System action:
Processing stops.

User response:
Define the missing objects.

ADBP013W **One or more related LOB objects have been defined. If you continue with choice number 2, those definitions will be removed.**

Explanation:
You requested that LOB objects be created implicitly. However, at least one LOB object is already explicitly defined for the LOB column. If you choose to continue, any explicit object definitions will be removed.

System action:
The **Add LOB Objects Confirmation (ADB2CONF)** panel is redisplayed with this warning message.

User response

Choose one of the options displayed:

1
Explicitly create (or re-create) definitions for the related LOB objects.

2
Remove the existing explicit definitions and have Db2 create them implicitly.

3
Cancel with no action.

ADBP014I **The LOB table space, auxiliary table, and auxiliary index for the new LOB columns will be created implicitly. If the table will be dropped and recreated, any explicit related objects for existing LOB columns will also become implicit. Press ENTER to continue.**

Explanation:
Existing LOB columns in the base table have a combination of explicitly created LOB objects and implicitly created LOB objects. In this case, for any new LOB columns, Db2 will implicitly create the related objects. In addition, if the table needs to be dropped and re-created, Db2 will implicitly create the related objects for all existing LOB columns.

System action:
Processing continues.

User response:
Press ENTER to continue with ALT processing.

ADBP015W **Changing the CCSID of a key column in a foreign key relationship requires that the CCSID of the corresponding key in the parent or child table match.**

Explanation:
This warning message is issued when the CCSID attribute is changed for a parent or foreign key column in a referential constraint. This could lead to a failure

during execution when ALT attempts to reestablish the foreign key.

System action:
Processing continues after you press Enter.

User response:
If possible, change the corresponding key in the parent or child table so that the attributes match. Otherwise, do not proceed with the CCSID attribute change.

ADBP016E **The schema name of the target sequence cannot begin with 'SYS' unless the schema name is 'SYSADM'.**

System action:
Processing stops.

User response:
Specify a schema name of a target sequence that does not start with SYS.

ADBP017E **The target sequence cannot be an existing sequence alias.**

System action:
Processing stops.

User response:
Specify a different sequence alias.

ADBP018E **The referenced object name cannot be the same as the alias name.**

Explanation:
The alias name cannot be the same as the table, view, or table alias name for which the alias is defined.

System action:
Processing stops.

User response:
Specify a referenced object name that is different than the alias name.

ADBP018W ***Reason* The object need not exist when the alias is defined, but it must exist when an SQL statement that contains the alias is used, otherwise an error is returned by Db2.**

Explanation

An object name was entered for either a nonexistent object or one that will be dropped during the change management apply process. Db2 allows an alias to be created for such an object, but doing so might produce an error when an SQL statement that contains the alias is used. In the message text, *reason* is one of the following phrases:

- The target sequence does not exist.
- A pending change to drop the target sequence exists.
- The referenced table or view does not exist.
- A pending change to drop the referenced table exists.

System action:
The panel is redisplayed with the warning message.

User response:
Either provide a new name for the reference object that is different than the alias, or press Enter to continue.

ADBP019E **Target sequence *sequence_name* for alias *alias_name* must exist in order to proceed.**

Explanation:
Although it is possible to create a sequence alias that refers to a target sequence that does not currently exist, it is not possible to use that alias in an SQL statement such as ALTER.

System action:
Processing stops.

User response:
To proceed, create the target sequence that the alias refers to.

ADBU000E **The UNLOAD utility does not support LOB table spaces.**

Explanation:
The DB2 UNLOAD utility will not process a LOB table space.

System action:
Processing stops.

User response:
Perform the unload on the base table space. The unload will contain the data from the LOB table space.

ADBU004W **Referential constraint has been detected where changing the CCSID field may introduce a mismatch between encoding schemes in parent and child tables.**

Explanation:
When the CCSID attribute is changed for a table space that's associated with a table that has referential constraint (RI Parent/Child table relationship), check the other table space and ensure the CCSID attribute is the same.

System action:
Processing continues.

User response:
Consider changing the CCSID attribute in the corresponding table space that has parent/child table

relationship with referential constraints so that the attributes match. Otherwise do not proceed with the CCSID attribute change.

ADBU012E For a partitioned table space, the Repair Utility with LEVELID option must be initiated at the partition level. Enter S in the line command field. Subsequently, enter SP in the line command field, then enter the utility dialog for the specific table space partition.

Explanation:
The REPAIR LEVELID utility cannot operate at the table space level. It must be initiated at the partition level.

System action:
The system waits.

User response:
Press F3 to return to the VIEW panel, then enter S by the view name. On the subsequent panel, entert SP for the table space that is shown. On the subsequent panel, enter the UTIL line command for the specific table space partition.

ADBU022E The PA template variable is required for a pending PAGENUM RELATIVE change.

Explanation:
A TEMPLATE must be allowed to allocate a separate data set for each partition because the table space uses relative page numbering. See also DB2 message DSNU2922I.

System action:
The system waits for the template data set to be changed.

User response:
Add the *&PA.* template variable.

ADBXF000 STOW error: a PDS directory is full or an I/O error occurred.

Explanation:
The specified PDS directory is full or an I/O error occurred.

System action:
Processing stops.

User response:
Ensure that the specified PDS directory is not full or fix the I/O error and try again.

ADBZ001E Table cannot be archived because *message*

Explanation

The table cannot be archived because *message*, where *message* is one of the following:

- no partitions were selected.
- no SYSACCELERATEDTABLES table exists.
- a table is not specified for accelerator.
- XML or LOB columns are present in the table.
- the table is a parent of foreign key relationship.
- the table is not in a partition by range table space.
- the stored procedure ACCEL_ARCHIVE_TABLE does not exist.

System action:
Processing ends.

User response:
If possible, fix any error conditions and try the operation again.

ADBZ002E Partition range is invalid because *message*

Explanation

The partition range is invalid because *message*, where *message* is one of the following:

- no spaces are allowed in the range list.
- an invalid character is in the range list.
- of invalid range list syntax.
- the ending part in the range construct must be greater.
- the part specified is greater than the maximum part.

System action:
Processing ends.

User response:
Change the partition range using valid syntax and partition values and try the operation again.

ADBZ009E The attempt to disable or enable incremental updates failed because *reason*.

Explanation

Disabling or enabling incremental updates failed because *reason*, where *reason* is one of the following:

- stored procedure ACCEL_SET_TABLES_REPLICATION does not exist.
- the associated accelerator is virtual.

System action:
Processing ends.

User response:

Ensure that the stored procedure exists and that the accelerator is not virtual and try the operation again.

ADBZ103I **The trace information has been successfully saved to the specified sequential data set.**

Explanation:
The specified trace information is collected.

System action:
Processing continues.

User response

No action is required.

ADBZ104E **Data set already exists.**

Explanation:
The specified sequential data set already exists. You must enter a unique value.

System action:
Processing stops.

User response:
If you want to reuse the data set, delete the existing data set and a new data set will be generated. If you want to create a new data set, change the modifier so that the data set fully qualified name is different.

ADBZ105E **Data set already exists.**

Explanation:
The specified sequential data set already exists. You must enter a unique value.

System action:
Processing stops.

User response:
If you want to reuse the data set, delete the existing data set and a new data set will be generated. If you want to create a new data set, change the modifier so that the data set fully qualified name is different.

BND001I **BIND MANAGER STARTED FOR DBRM=*dbrmname***

Explanation

The Bind Manager function of Db2 Admin Tool is initializing.

dbrmname
The name of the DBRM that is being processed.

Module: ADBBMAV

System action:
Processing continues.

User response

No action is required.

BND002I **OLD DBRM INFORMATION:TIMESTAMP: CHAR=*yyyy.mm.dd.hh.mm.ss.xxxx* HEX= *hhhhhhhh hhhhhhhh* DECIMAL=*dddddddd dddddddd* USERID=*userid* VERSION=*version* CCSID=*ccsid***

Explanation

The preexisting DBRM was located and processed successfully.

yyyy.mm.dd.hh.mm.ss.xxxx
The timestamp in the DBRM.

hhhhhhhh hhhhhhhh
The DBRM timestamp in hexadecimal format.

dddddddd dddddddd
The DBRM timestamp in decimal format.

userid
The user ID of the last user to precompile the program.

version
The version identifier for the DBRM. This identifier can be up to 64 characters long.

The version information is included only if the DBRM was created with the VERSION parameter.

ccsid
The CCISD of the DBRM.

Module: ADBBMAV

System action:
Processing continues.

User response

No action is required.

BND003I **NEW DBRM INFORMATION:TIMESTAMP: CHAR=*yyyy.mm.dd.hh.mm.ss.xxxx* HEX= *hhhhhhhh hhhhhhhh* DECIMAL=*dddddddd dddddddd* USERID=*userid* VERSION=*version* CCSID=*ccsid***

Explanation

A new DBRM was successfully created by the Db2 precompiler

yyyy.mm.dd.hh.mm.ss.xxxx
The timestamp in the new DBRM.

hhhhhhhh hhhhhhhh
The DBRM timestamp in hexadecimal format.

dddddddd dddddddd
The DBRM timestamp in decimal format.

userid
The user ID of the current user who is running the job.

version
The version identifier for the DBRM. This identifier can be up to 64 characters long.

The version information is included only if the DBRM was created with the VERSION parameter.

ccsid
The CCISD of the DBRM.

Module: ADBBMAV

System action:
Processing continues.

User response

No action is required.

BND004I **PROCESSING COMPLETED SUCCESSFULLY, *** BIND NOT REQUIRED *****

Explanation

Bind Manager completed processing and determined that a bind is not required. The existing DBRM and subsequent source updates are compatible. You can process the load module without a Db2 bind.

Module: ADBBMAV

System action:
Processing stops.

User response:
If the bind step tests for a return code from Bind Manager, no action is required. Otherwise, omit the step of binding a new DBRM.

Severity

0 (informational)

BND005I **DBRM COMPARE FAILED, SOURCE MODIFICATIONS HAVE CHANGED THE CONTENT OF THE DBRM**

Explanation

Updates to the source code changed the SQL structure. A Db2 bind is required. This message is followed by a dump of both the existing DBRM records and new DBRM records that do not match.

Module: ADBBMAV

System action:
Processing continues.

User response:
If the bind step tests for a return code from Bind Manager, no action is required. Otherwise, bind the new DBRM and copy it to the DBRM library.

BND006E ***ddname*** **DCB FAILED TO OPEN**

Explanation

Bind Manager could not open the specified data set.

ddname
The DD statement that defines the data set.

Module: ADBBMAV

System action:
Processing stops.

User response:
Correct the DD statement and resubmit the job.

BND007E ***sss*** **TIME STAMP HAS INVALID FORMAT, "LEVEL" PRECOMPILER OPTION NOT SUPPORTED**

Explanation

The specified DBRM was generated with the Db2 SQL processing option LEVEL. This option is not supported by Bind Manager.

sss
Identifies the DBRM. Possible values are OLD or NEW.

Module: ADBBMAV

System action:
Processing stops.

User response

If *sss* is OLD, delete the old DBRM from the library that is specified by the DBRMLIB DD statement.

If *sss* is NEW, remove the LEVEL option from the PARM parameter of the EXEC statement.

Related reference

Descriptions of SQL processing options (Db2 12 for z/OS)

BND008E **LANGUAGE TYPE *type* IS NOT SUPPORTED**

Explanation

The Db2 SQL processing option HOST specified a language that is not supported by Bind Manager.

type
The specified host language.

Supported language types are COB2, COBOL, PLI, and ASM.

Module: ADBBMAV

System action:
Processing stops.

User response:
Correct the HOST option and resubmit the job.

Related reference

Descriptions of SQL processing options (Db2 12 for z/OS)

BND010S **INTERNAL ERROR DURING DECODE OR UPDATE OF THE NEW DBRM**

Explanation

Bind Manager has encountered an internal processing error when updating the DBRM. The contents of the new DBRM are unpredictable.

Module: ADBBMAV

System action:
Processing stops.

User response:
Verify that DBRMLIB DD statement specifies a valid library and member name and that the old DBRM contains valid data. If this action does not resolve the problem, save the dump from the SYSUDUMP DD statement and contact IBM Software Support.

BND011E **DBRM EXCEEDS THE 3,000 SQL CALL RESTRICTION**

Explanation

Bind Manager detected a DBRM that contains more than 3,000 SQL CALL statements. This number exceeds the maximum supported program size.

Module: ADBBMAV

System action:
Processing stops.

User response:
Restructure the application to reduce the number of SQL calls.

BND012E **SYNTAX ERROR OR UNRECOGNIZABLE OPERAND IN THE PRECOMPILER PARAMETER**

Explanation

Bind Manager detected a syntax error in the PARM parameter of the EXEC statement for the Db2 precompiler.

Module: ADBBMAV

System action:
Processing stops.

User response:
Correct the error and resubmit the job.

BND013E **UNABLE TO ALLOCATE THE TEMPORARY DBRM SAVE DATASET**

Explanation

Bind Manager could not allocate a temporary DASD data set for its work file.

Module: ADBBMAV

System action:
Processing stops.

User response:
Ensure that enough DASD work space is available and resubmit the job.

BND014E **RETURN CODE FROM THE SQL PRECOMPILER > 4**

Explanation

The Db2 precompiler ended with a return code that is greater than four.

Module: ADBBMAV

System action:
Processing stops.

User response:
Correct the SQL errors in the source code and resubmit the job.

BND015E **DBRMLIB DATASET HAS AN LRECL OTHER THAN 80**

Explanation

The data set that is specified in the DBRMLIB DD statement does not have the correct record length. Bind Manager and the Db2 precompiler require that this data set have a record length of 80 bytes (LRECL=80).

Module: ADBBMAV

System action:
Processing stops.

User response:
Verify that the DBRMLIB DD statement points to the correct existing DBRM and the library has the correct attributes (RECFM=FB and LRECL=80), and resubmit the job. If this action does not resolve the problem, contact IBM Software Support.

BND016E **MEMBER POINTED TO BY THE DBRMLIB DD IS NOT A VALID DBRM**

Explanation

Bind Manager detected an invalid DBRM. Likely, the DBRMLIB DD statement does not specify a valid DBRM library.

Module: ADBBMAV

System action:
Processing stops.

User response:
Verify that the DBRMLIB DD statement references the correct DBRM library.

BND017E **UNABLE TO FIND OLD DBRM MEMBER IN DBRMLIB, *dbrm-name* IS ASSUMED TO BE NEW**

Explanation

Bind Manager could not find the DBRM member in the library that is specified in the DBRMLIB DD statement. This DBRM is not processed.

dbrmname
The name of the DBRM that cannot be found.

Module: ADBBMAV

System action:
Processing continues.

User response:
Verify that the DBRMLIB DD statement references the correct DBRM library.

BND018S **BIND MANAGER HAS DETECTED AN S*xxx* ABEND, ATTEMPTING RECOVERY**

Explanation

An abend occurred during processing. The Bind Manager ESTAE routine is attempting to recover and continue.

S*xxx*
The system abend completion code.

Module: ADBBMAV

System action:
Processing continues.

User response:
If recovery is successful and the job completes normally, no action is required. Otherwise, contact IBM Software Support.

BND019E **Deletion of source statements has invalidated the DBRM—add comment lines to correct // BND019E Old stmt nbr=xxxx (hex), New stmt nbr=yyyy (hex)**

Explanation

Statements were deleted from a large source module.

With the Db2 precompiler, when a COBOL program grows so that at least one SQL statement has a relative line number greater than 9999, the internal format of the DBRM changes. This changed internal format can later cause a compatibility problem if lines are subsequently deleted so that all SQL statements again have line numbers less than 10,000. This sequence of events creates a condition that Bind Manager cannot handle. Therefore, a bind cannot be avoided.

The situation does not arise in DB2 UDB for z/OS 8 new-function mode or later.

xxxx* and *yyyy
The statement numbers from the DBRM records in which the condition was detected

Module: ADBBMAV

System action:
Processing continues.

User response

If the SQL structure was not changed, you can circumvent this situation by taking the following steps:

1. Restore the old DBRM (the version that was saved in the job step before the precompiler step).
2. Rather than deleting source statements, comment them out. This action leaves the relative line numbers unchanged.
3. Rerun the compile job.

BND020E **DB2 VERSION/RELEASE MISMATCH BETWEEN OLD/NEW DBRMS**

Explanation

The old and new DBRMs were created under different versions of Db2.

Module: ADBBMAV

System action:
Processing stops.

User response

You must do an initial bind under the new version; afterwards Bind Manager functions normally with the new DBRM format.

Recommendation: To allow for possible modifications to the optimizer, always bind existing applications when the Db2 version changes.

BND021E **Unsupported release of DB2**

Explanation

The DBRM was created by an unrecognized release of Db2.

Module: ADBBMAV

System action:
Processing stops.

User response:
Bind the DBRM on a supported release of Db2.

BND022E DBRM COMPARE FAILED—NONMATCHING CHARACTER SETS (CCSIDs)

Explanation

A mismatch was detected between the old and the new DBRMs, because they were created with different coded character set identifiers (CCSIDs). A bind is required.

Module: ADBBMAV

System action:
Processing continues.

User response:
If the bind step tests for a return code from Bind Manger, no action is required. Otherwise, bind the new DBRM and copy it to the DBRM library.

BND023S DB2 BIND MANAGER PROCESSING TERMINATED DUE TO UNRECOVERABLE ERROR

Explanation

Bind Manager encountered a severe error and cannot continue. This message is normally preceded by another message that states the specific error or cause of failure.

Module: ADBBMAV

System action:
Processing continues.

User response

Save all output. Gather the following materials and contact IBM Software Support

- the preexisting (old) DBRM
- the new DBRM, if one was created
- all SYSOUT from the job that failed (the entire job, not just the Bind Manager job step)

BND024S ** ERROR ** SSID DD STATEMENT IS PRESENT BUT NO SUBSYSTEM WAS SPECIFIED

Explanation

A DBRM was requested by including an SSID DD statement in the JCL, but no subsystem name was specified. Therefore, Bind Manager does not know which Db2 system catalog to use for generating the DBRM.

Module: ADBBMAV

System action:
Processing stops.

User response

Add the required SSID DD statement to the JCL as follows:

```
//SSIDxyz DD DUMMY
```

xyz is the 1- to 4-character subsystem ID.

For example, the following DD statement requests that Bind Manager use subsystem DSN1 to generate the DBRM:

```
 //SSIDDSN1 DD DUMMY
```

BND025I UNABLE TO OPEN SYSPRINT UNABLE TO EXTRACT DBRM FROM CATALOG SYMPTOMS= *xxxxxxxx/ yyyyyyyy*

Explanation

A DBRM was requested, but Bind Manager could not extract the relevant information from the catalog. This error normally occurs when the program is being compiled for the first time, because the catalog does not contain an existing copy of the DBRM. A bind is required.

xxxxxxxx
Return code

yyyyyyyy
Reason code

Module: ADBBMAV

System action:
Processing continues.

User response:
Because this condition is usually not an error, no action is required. However, if you are certain that the catalog contains a valid DBRM for the program that is being compiled, save all output and contact IBM Software Support.

BND026I THE "OLD" DBRM WAS FOUND IN PLAN *plan-name* IN THE DB2 CATALOG

Explanation

A DBRM was generated based on the DBRM in the specified plan.

plan-name
The name of the plan.

Module: ADBBMAV

System action:
Processing continues.

User response

No action is required.

BND027I THE "OLD" DBRM WAS FOUND IN VERSION *version* IN THE DB2 CATALOG

Explanation

A DBRM was generated based on the specified package version.

version
The version identifier.

Module: ADBBMAV

System action:
Processing continues.

User response

No action is required.

BND028I DBRM NOT FOUND IN CATALOG

Explanation

Bind Manager was unable to read the old DBRM.

Module: ADBBMAV

System action:
Processing stops.

User response:
See message BND098I or BND099S for more information.

Related information

"BND098I" on page 1203
BIND MANAGER PROCESSING TERMINATED, !! *reason* !!

"BND099S" on page 1203
BIND MANAGER PROCESSING FAILED DURING DBRM UPDATE, DELETE THE DBRM AND RERUN

BND029W CCSID check bypassed by user request

Explanation

Because the NOCC DD statement was specified, CCSID checking was suppressed.

Module: ADBBMAV

System action:
Processing continues.

User response

No action is required.

If you do not want this behavior, remove the NOCC DD statement.

BND030W BNDWK1 DD STATEMENT MISSING, DBRM NOT GENERATED

Explanation

The required BNDWK1 DD statement is missing.

Module: ADBBMAV

System action:
Processing continues.

User response:
Add a BNDWK1 DD statement to the JCL and resubmit the job.

BND031E NEW DBRM IS NOT VALID

Explanation

The DBRM that was produced by this precompile operation is damaged or corrupted. Specifically, the DBRM string, or "eye-catcher" text, is not included where it should be.

Module: ADBBMAV

System action:
Processing stops.

User response:
Contact IBM Software Support.

BND032S VAL DD statement is present but no subsystem ID was specified

Explanation

VAL is an optional parameter that indicates that the bind avoidance program is to check for invalid plans or packages. A subsystem ID must be specified when using this option.

Module: ADBBMAV

System action:
Processing stops.

User response

Omit the VAL DD statement or add the required SSID as follows:

```
//VAL_xyz DD Dummy
```

where _*xyz* is the 1- to 4-character subsystem ID.

BND033S **SSID and VAL DD statements are mutually exclusive; only one may be specified**

Explanation

Either an SSID or VAL DD statement can be specified. You cannot specify both.

Module: ADBBMAV

System action:
Processing stops.

User response:
Omit either the SSID or VAL DD statement.

BND034I **At least one plan or package containing this DBRM is invalid**

Explanation

The bind avoidance program detected an invalid DBRM or DBRMs.

Module: ADBBMAV

System action:
Processing continues.

User response

No action is required.

BND035I **The following plans are potentially affected and should be bound:**

Explanation

The bind avoidance program detected a plan or plans that need to be bound.

Module: ADBBMAV

System action:
Processing continues.

User response:
If appropriate, bind the plan. Otherwise, no action is required.

BND036I **This package is potentially affected in the following collections and should be bound**

Explanation

The bind avoidance program detected a package that should be bound.

Module: ADBBMAV

System action:
Processing continues.

User response:
If appropriate, bind the package. Otherwise, no action is required.

BND037E **Unable to connect to subsystem *ssss* RC=*xxxxxxxx* Reason=*yyyyyyyy***

Explanation

Bind manager was unable to connect to specified Db2 subsystem.

ssss
The subsystem ID

xxxxxxxx
Return code from DSNALI

yyyyyyyy
Reason code from DSNALI

Module: ADBBMAV

System action:
Processing stops.

User response

Use a valid Db2 subsystem ID in the following DD statement:

```
//SSIDssss DD DUMMY
```

BND038W **Unicode conversion failed CC=*xxxxxxxx*, RC=*yyyyyyyy*, REASON=*zzzzzzzz***

Explanation

Bind Manager was unable to convert the DBRM from Unicode format.

xxxxxxxx
Return code from ADBBMUN

yyyyyyyy
Return code from CUNLCNV

zzzzzzzz
Reason code from CUNLCNV

Module: ADBBMAV

System action:
Unicode conversion is skipped, and processing continues.

User response

No action is required.

However, if the DBRM must be converted from Unicode, save the output and contact IBM Software Support.

BND039W Load failed for Unicode conversion module CUNLCNV

Explanation

Bind Manager was unable to load module CUNLCNV.

Module: ADBBMAV

System action:
Unicode conversion is skipped, and processing continues .

User response

No action is required.

However, if the DBRM must be converted from Unicode, save the output and contact IBM Software Support.

BND040W Unicode Initialization failed CC=*xxxxxxxx*, RC=*yyyyyyyy*, REASON=*zzzzzzzz*

Explanation

Bind Manager was unable to initialize Unicode conversion module CUNLCNV.

xxxxxxxx
Return code from ADBBMUN

yyyyyyyy
Return code from CUNLCNV

zzzzzzzz
Reason code from CUNLCNV

Module: ADBBMAV

System action:
Unicode conversion is skipped, and processing continues.

User response

No action is required.

However, if the DBRM must be converted from Unicode, save the output and contact IBM Software Support.

BND098I BIND MANAGER PROCESSING TERMINATED, !! *reason* !!

Explanation

Bind Manager processing ended for the given reason.

reason
The reason that processing terminated:

BIND REQUIRED
The process ended, because it determined that a Db2 bind is required.

CORRECT PRECOMPILER ERRORS
The process ended because of Db2 precompiler errors. In this case, this message is preceded by message BND014E.

This message is normally accompanied by return code 4.

Module: ADBBMAV

System action:
Processing continues.

User response:
If *reason* is BIND REQUIRED, and the bind procedure tests for a return code from Bind Manager, no action is required. Otherwise, fix the source program and resubmit the job or bind the new DBRM and copy it to the DBRM library.

Related information

"BND014E" on page 1198
RETURN CODE FROM THE SQL PRECOMPILER > 4

BND099S BIND MANAGER PROCESSING FAILED DURING DBRM UPDATE, DELETE THE DBRM AND RERUN

Explanation

Bind Manager failed during the DBRM update. This message is accompanied by abend code U999 and a dump of user areas (as directed by the SYSUDUMP DD statement). The DBRM might be corrupted.

Module: ADBBMAV

System action:
Processing stops.

User response:
Ensure that the DBMRLIB DD statement specifies the correct library and that the user who submitted the job has authority to update the library. If the problem persists, contact IBM Software Support.

BND2101S UNABLE TO CONNECT TO SUBSYSTEM *subsystem*, PLAN ADBBMRG RC=*return code*, REASON=*reason code*

Explanation

An unexpected Db2 error occurred while the program was trying to connect to the specified subsystem.

subsystem
The subsystem ID.

return code
The return code from Db2.

reason code
The reason code from Db2.

Module: ADBBMRG

System action:
Processing stops.

User response:
Ensure that you specified the correct subsystem ID and that the subsystem is up and available. Then, resubmit the job.

BND2102S UNABLE TO OPEN *ddname*, REASON=*reason code*

Explanation

The program could not open the specified data set for the given reason.

ddname
The DD statement that identifies the data set.

reason code
The reason code.

Module: ADBBMRG

System action:
Processing stops.

User response:
Verify that the DD statement was specified correctly. If so, contact IBM Software Support.

BND2104I *text*

Explanation

This informational message displays the input that was read from SYSIN.

text
The input command or comment from SYSIN.

Module: ADBBMRG

System action:
Processing continues.

User response

No action is required.

BND2105E INVALID STATEMENT

Explanation

The input statement that precedes this message is invalid.

Module: ADBBMRG

System action:
Processing stops.

User response:
Correct or remove the invalid statement and resubmit the job.

BND2106E MISSING OR INVALID SUBSYSTEM ID (SSID)

Explanation

The SSID command does not specify a valid subsystem ID.

Module: ADBBMRG

System action:
Processing stops.

User response:
Correct the command and resubmit the job.

BND2107I CONNECTED TO SUBSYSTEM *subsystem*

Explanation

A connection was established to the specified subsystem.

subsystem
The subsystem ID.

Module: ADBBMRG

System action:
Processing continues.

User response

No action is required.

BND2108E SSID MAY ONLY BE SPECIFIED ONCE

Explanation

More than one SSID command was specified. You can specify only one SSID command each time that you run Bind Manager.

Module: ADBBMRG

System action:
Processing stops.

User response:
Remove the extra SSID commands and resubmit the job.

BND2110E NO SSID WAS SPECIFIED

Explanation

A valid SSID command was not specified. The program cannot to connect to Db2.

Module: ADBBMRG

System action:
Processing stops.

User response:
Add an SSID command and resubmit the job.

BND2116W ERROR IN PACKAGE STATEMENT: MISSING OPERAND(S)

Explanation

The PACKAGE command does not include any operands.

Module: ADBBMRG

System action:
Processing continues.

User response:
Correct the command and resubmit the job.

BND2121S MODULE DSNTIAR COULD NOT BE LOADED, REASON=*reason code*

Explanation

DSNTIAR could not be accessed.

reason code
The reason code.

Module: ADBBMRG

System action:
Processing stops.

User response:
Verify that DSNTIAR is accessible, either through a STEPLIB DD statement or the system link list.

BND2122S DSNTIAR FAILURE, RC=*return code*

Explanation

DSNTIAR failed with the specified return code.

return code
The return code.

Module: ADBBMRG

System action:
Processing stops.

User response:
Verify that DSNTIAR is accessible, either through a STEPLIB DD statement or the system link list. Also verify that you are calling the correct version of DSNTIAR for the Db2 subsystem that is being accessed.

BND2125S UNABLE TO DISCONNECT FROM SUBSYSTEM *subsystem* RC=*return code*, REASON=*reason code*

Explanation

An unexpected Db2 error occurred when the program tried to disconnect from the specified subsystem.

subsystem
The subsystem ID.

return code
The return code from Db2.

reason code
The reason code from Db2.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify the correct subsystem ID, ensure that the subsystem is up and available, and resubmit the job.

BND2126I DISCONNECTED FROM SUBSYSTEM *subsystem*

Explanation

The connection to the specified subsystem is broken.

subsystem
The subsystem ID.

Module: ADBBMRG

System action:
Processing continues.

User response

No action is required.

BND2129I MORE THAN ONE VERSION OF PACKAGE *package* EXISTS; SELECT A SPECIFIC VERSION TO PROCESS

Explanation

The PACKAGE command did not specify a version, but more than one version of the package exists in the subsystem.

package
The name of the package.

Module: ADBBMRG

System action:
Processing continues.

User response:
Specify a valid version of the package.

BND2130I **PACKAGE *package* NOT FOUND IN SUBSYSTEM *subsystem***

Explanation

The named package does not exist in the specified subsystem.

package
The name of the package.

subsystem
The subsystem ID.

Module: ADBBMRG

System action:
Processing continues.

User response:
Specify an existing package.

BND2132W **PACKAGE NOT FOUND IN SUBSYSTEM *subsystem***

Explanation

The specified package version does not exist in the specified collection on the named Db2 subsystem.

subsystem
The subsystem ID.

Module: ADBBMRG

System action:
Processing continues.

User response:
Ensure that the combination of values that are specified for package, version, and collection identify an existing package.

BND2133I ***dbrm* EXISTS IN MORE THAN ONE PLAN; SELECT A SPECIFIC PLAN TO PROCESS**

Explanation

More than one instance of the specified DBRM exists.

dbrm
The name of the DBRM.

Module: ADBBMRG

System action:
Processing continues.

User response:
Identify a specific DBRM instance by specifying a plan name.

BND2134I **DBRM *dbrm* NOT FOUND IN SUBSYSTEM *subsystem***

Explanation

The named DBRM does not exist on the subsystem.

dbrm
The name of the DBRM.

subsystem
The subsystem ID.

Module: ADBBMRG

System action:
Processing continues.

User response:
Specify an existing DBRM.

BND2135E **DBRM NOT FOUND IN SUBSYSTEM *subsystem***

Explanation

The specified DBRM does not exist in the specified plan on the named Db2 subsystem.

subsystem
The subsystem ID.

Module: ADBBMRG

System action:
Processing stops.

User response:
Ensure that the values specified for plan and DBRM identify an existing DBRM.

BND2136E **DBRM GENERATION FAILED, REASON=*reason code***

Explanation

An internal processing error has occurred. Bind Manager is attempting to extract DBRM information from the Db2 catalog.

reason code
The reason code.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify the correct subsystem name, ensure that the subsystem is up and available, and resubmit the job.

BND2137I **DBRM *dbrm* CREATED**

Explanation

The DBRM was generated and stored in the library that is identified by the DBRMLIB DD statement.

dbrm
The name of the DBRM.

Module: ADBBMRG

System action:
Processing continues.

User response

No action is required.

BND2138E PACKAGE GENERATION FAILED, REASON=*reason code*

Explanation

An internal processing error occurred.

reason code
The reason code.

Module: ADBBMRG

System action:
Processing stops.

User response:
Contact IBM Software Support.

BND2139I PACKAGE *package* CREATED

Explanation

The package has been re-created as a DBRM and stored in the library that is identified by the DBRMLIB DD statement.

package
The name of the package.

Module: ADBBMRG

System action:
Processing continues.

User response

No action is required.

BND2142E *ddname* DD STATEMENT MISSING

Explanation

The specified DD statement is required.

ddname
The DD statement name.

Module: ADBBMRG

System action:
Processing stops.

User response:
Supply the missing DD statement and resubmit the job.

BND2143E UNABLE TO OPEN DBRMLIB, REASON=*reason code*

Explanation

The data set that is identified by the DBRMLIB DD statement could not be opened.

reason code
The reason code.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify a valid DBRMLIB DD statement. Ensure that the DD statement refers to a valid library and that the library has the correct attributes. For example, the data set should have the attributes RECFM=FB and LRECL=80.

BND2144I DBRM *dbrm* REPLACED

Explanation

A new DBRM was generated. This DBRM replaces the existing DBRM in the library that is identified by the DBRMLIB DD statement.

dbrm
The name of the DBRM that was generated.

Module: ADBBMRG

System action:
Processing continues.

User response

No action is required.

BND2145I PACKAGE *package* REPLACED

Explanation

The package was re-created as a DBRM. This DBRM replaces the existing DBRM in the library that is identified by the DBRMLIB DD statement.

package
The name of the package that was re-created.

Module: ADBBMRG

System action:
Processing continues.

User response

No action is required.

BND2146S BPAM API FAILURE, FUNCTION=WRITE, SYMPTOM=*symptom*

Explanation

A basic partitioned access method (BPAM) write failure occurred.

symptom
The symptom of the failure.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify a valid DBRMLIB DD statement. Ensure that the DD statement refers to a valid library and that the library has the correct attributes. For example, the data set should have the attributes RECFM=FB and LRECL=80.

BND2147S BPAM API FAILURE, FUNCTION=WRITE, SYMPTOM=*symptom*

Explanation

A failure occurred when writing out the residual buffer.

symptom
The symptom of the failure.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify a valid DBRMLIB DD statement. Ensure that the DD statement refers to a valid library and that the library has the correct attributes. For example, the data set should have the attributes RECFM=FB and LRECL=80.

BND2148S BPAM API failure, function=STOW, symptom=*symptom*

Explanation

A STOW failure occurred when attempting to write the DBRM member name to a partitioned data set (PDS).

symptom
The STOW return code

Module: ADBBMRG

System action:
Processing stops.

User response:
Read the STOW return code information to determine the action to take.

Related information

STOW completion codes (z/OS 3.1.0)

BND2149E *keyword* IS NOT A VALID KEYWORD FOR THE LIST COMMAND

Explanation

The LIST command has a syntax error.

keyword
The specified keyword that is not valid.

Module: ADBBMRG

System action:
Processing stops.

User response:
Correct the command and resubmit the job.

BND2150E DEFAULTS MAY ONLY BE SPECIFIED ONCE

Explanation

Multiple DEFAULTS commands were specified.

Module: ADBBMRG

System action:
Processing stops.

User response:
Remove all but one of the DEFAULTS commands and resubmit the job.

BND2151W NO OPERANDS SPECIFIED ON DEFAULTS STATEMENT

Explanation

No operands were specified on the DEFAULTS statement.

Module: ADBBMRG

System action:
Processing continues.

User response:
Specify valid operands or remove the command.

BND2152W DEFAULTS STATEMENT HAS INVALID SYNTAX

Explanation

The syntax of the DEFAULTS statement is invalid because of multiple commas.

Module: ADBBMRG

System action:
Processing continues.

User response:
Correct the command or remove it.

BND2153W **DEFAULTS STATEMENT HAS INVALID SYNTAX**

Explanation

The syntax of the DEFAULTS statement is invalid because of misplaced commas.

Module: ADBBMRG

System action:
Processing continues.

User response:
Correct the command or remove it.

BND2156W ***operand* IS NOT A VALID OPERAND FOR THIS STATEMENT. THE STATEMENT IS IGNORED.**

Explanation

The second operand is not valid for the DEFAULTS command.

operand
The invalid operand.

Module: ADBBMRG

System action:
Processing continues.

User response:
Correct the invalid operand or remove the command.

BND2157W **DEFAULTS STATEMENT DOES NOT SPECIFY A VALID LANGUAGE. THE STATEMENT IS IGNORED.**

Explanation

The language that was specified in the DEFAULTS statement is invalid.

Module: ADBBMRG

System action:
Processing continues.

User response:
Specify a valid language or remove the command.

BND2158W **DEFAULTS STATEMENT DOES NOT SPECIFY A VALID DB2 VERSION. THE STATEMENT IS IGNORED**

Explanation

The Db2 version that was specified in the DEFAULTS statement is invalid.

Module: ADBBMRG

System action:
Processing continues.

User response:
Specify a valid version or remove the command.

BND2159W ***operand* IS NOT A VALID OPERAND FOR THIS STATEMENT. THE STATEMENT IS IGNORED.**

Explanation

The second operand is not valid for the DEFAULTS command.

operand
The invalid operand.

Module: ADBBMRG

System action:
Processing continues.

User response:
Correct the invalid operand or remove the command.

BND2160W **The first operand *operand* IS INVALID. THE STATEMENT IS IGNORED.**

Explanation

The first operand is not valid for the DEFAULTS command.

operand
The invalid operand.

Module: ADBBMRG

System action:
Processing continues.

User response:
Correct the invalid operand or remove the command.

BND2161W **WARNING - ONE OR MORE GENERATED DBRMS ARE MISSING DB2 VERSION AND/OR HOST LANGUAGE VALUES, AND NO DEFAULT VALUES WERE SUPPLIED.**

Explanation

DBRMs were generated, but they might not be valid, because they do not include a Db2 version value, a host language value, or both. A DEFAULTS command was not specified to supply the missing values.

Module: ADBBMRG

System action:
Processing stops.

User response

On the **Regenerate a DBRM Member (ADBPBMRD)** panel, specify a value for **Db2 version**, **Language**, or both to generate the DEFAULTS control statement and resubmit the job.

BND2162E NO DB2 VERSION WAS SPECIFIED.

Explanation

In the DEFAULTS command, a value was not specified for the DB2 keyword.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify a value for DB2 and resubmit the job.

BND2163E NO LANGUAGE WAS SPECIFIED

Explanation

In the DEFAULTS command, a value was not specified for the LANGUAGE keyword.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify a value for LANGUAGE and resubmit the job.

BND2164E ERROR IN DBRM STATEMENT: MISSING OPERAND(S)

Explanation

A DBRM command was specified with no operands.

Module: ADBBMRG

System action:
Processing stops.

User response:
Correct the DBRM command and resubmit the job.

BND2165E ERROR IN DBRM STATEMENT: MISSING DBRM NAME

Explanation

A DBRM command was specified with a plan name but without a DBRM name.

Module: ADBBMRG

System action:
Processing stops.

User response:
Correct the DBRM command to include a DBRM name and resubmit the job.

BND2166E ERROR IN PACKAGE STATEMENT: MISSING PACKAGE NAME

Explanation

A PACKAGE command was specified with a collection name but without a package name.

Module: ADBBMRG

System action:
Processing stops.

User response:
Correct the PACKAGE command to include a package name and resubmit the job.

BND2167E ERROR IN PACKAGE STATEMENT: MISSING VERSION

Explanation

A PACKAGE command was specified with a collection name and package name, but without a version ID.

Module: ADBBMRG

System action:
Processing stops.

User response:
Correct the PACKAGE command to include a version ID and resubmit the job.

BND2168E ERROR IN PACKAGE STATEMENT: SYNTAX -- VERSION

Explanation

A PACKAGE command was specified with an invalid version ID.

Module: ADBBMRG

System action:
Processing stops.

User response:
Correct the version ID in the PACKAGE command and resubmit the job.

BND2171S INTERNAL PROCESSING ERROR -- CALL TECHNICAL SUPPORT

Explanation

A program logic error occurred.

Module: ADBBMRG

System action:
Processing stops.

User response:
Save all output from the job, not just the job step, and contact IBM Software Support.

BND2172S UNEXPECTED END-OF-DATA ENCOUNTERED READING DBRM TEXT

Explanation

An end-of-data condition was reached prematurely while reading DBRM data from the Db2 catalog.

Module: ADBBMRG

System action:
Processing stops.

User response:
Contact IBM Software Support.

BND2173W **WARNING: UNABLE TO REFORMAT HOST VARIABLE PLACEHOLDER IN STATEMENT *statement*, SECTION *section*, AT COLUMN *column***

Explanation

Bind Manager could not insert the optimal number of blanks into the generated SQL statement.

statement
The generated SQL statement.

section
The SQL section number.

column
The column number. This value indicates the location where the blanks could not be inserted.

Module: ADBBMRG

System action:
Processing continues.

User response

No action is required.

BND2174I **No DBRMs found that match the specified name or mask**

Explanation

Bind Manager did not find any DBRMs that match the input criteria.

Module: ADBBMRG

System action:
Processing stops.

User response:
Check the input DBRMs and correct the input criteria as needed.

BND2175I **No packages found that match the specified name or mask**

Explanation

Bind Manager did not find any packages that match the input criteria.

Module: ADBBMRG

System action:
Processing stops.

User response:
Check the input packages and correct the input criteria as needed.

BND2176E **Invalid remote subsystem ID**

Explanation

The remote subsystem ID (SSID) is invalid.

Module: ADBBMRG

System action:
Processing stops.

User response:
Correct the remote SSID and resubmit the job.

BND2177E **DB2 RELEASES EARLIER THAN 2.3 ARE NOT SUPPORTED; *package* WAS NOT GENERATED.**

Explanation

You requested that Bind Manager generate a DBRM for a package or plan that was created for Db2 for z/OS Version 2.2 or earlier. Bind Manager does not support DBRM generation for those versions of Db2. The specified DBRM was not generated.

package
The name of the package or plan for which the DBRM was not generated.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify a valid package name.

BND2178E **Unable to connect to subsystem *ssid* RC= *return code*, REASON= *reason code***

Explanation

An unexpected Db2 error occurred while ADBBMRG tried to connect to the specified subsystem.

ssid
The subsystem ID.

return code
The return code from Db2.

reason code
The reason code from Db2.

Module: ADBBMRG

System action:

Processing stops.

User response:
Ensure that the subsystem is up and available.

BND2179E **Invalid package type *package type*; Package *package name* was not generated**

Explanation

Only packages that are created by the BIND PACKAGE command can be used by Bind Manager to generate a DBRM. However, the specified package was not created by the BIND PACKAGE command, as indicated by the package type in the Db2 catalog table SYSIBM.SYSPACKAGE.

package type
The package type, as recorded in the TYPE column of SYSIBM.SYSPACKAGE. (If this package was created by the BIND PACKAGE command, the TYPE column would be blank.)

package name
The name of the package.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify a valid package name.

BND2180E **Invalid Consistency Token HTTPREST; Package *package name* was not generated**

Explanation

The consistency token indicates that this package was created by a Db2 REST service. This type of package is not currently supported for DBRM regeneration.

package name
The package name.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify a valid package name.

BND2181E **Unable to identify DB2 release; *package* was not generated.**

Explanation

Bind Manager was unable to find a Db2 for z/OS release number for this package. The specified DBRM was not generated.

package
The name of the package for which the DBRM was not generated.

Module: ADBBMRG

System action:
Processing stops.

User response:
Specify a package with a valid Db2 release value.

BND2199S **ERROR IN ADBBMRG - SYMPTOM=*symptom***

Explanation

Bind Manager is attempting to display a nonexistent message. This condition is a program logic error.

symptom
The symptom of the failure.

Module: ADBBMRG

System action:
Processing stops.

User response:
Save all output and contact IBM Software Support.

BND2401S ***ddname* DD STATEMENT MISSING**

Explanation

The required DD statement was not included in the JCL.

ddname
The DD statement name.

Module: ADBBMCL

System action:
Processing stops.

User response:
Add the missing DD statement and resubmit the job.

BND2402S **UNABLE TO OPEN *ddname*, REASON=*reason code***

Explanation

The specified data set could not be opened.

ddname
The DD statement that defines the data set.

reason code
The reason for the failure. For an explanation of the reason code, see Enterprise PL/I for z/OS Messages and Codes (Enterprise PL/I for z/OS 5.3).

Module: ADBBMCL

System action:
Processing stops.

User response:
Correct the condition that is preventing the data set from being opened and resubmit the job.

BND2403S **MODULE *module* COULD NOT BE LOADED, REASON=*reason code***

Explanation

The z/OS FETCH function failed for the program load module. A common cause of this error is a missing STEPLIB statement in the JCL.

module
The load module for the program.

reason code
The reason for the failure. For an explanation of the reason code, see Enterprise PL/I for z/OS Messages and Codes (Enterprise PL/I for z/OS 5.3).

Module: ADBBMCL

System action:
Processing stops.

User response:
Correct the condition that caused the FETCH error and resubmit the job.

BND2404E **NO SUBSYSTEM NAME HAS BEEN SPECIFIED**

Explanation

A Db2 subsystem name is required in this situation so that Bind Manager knows which Db2 subsystem to access. However, a subsystem name was not specified.

Module: ADBBMCL

System action:
Processing stops.

User response:
Add the SSID command with a valid subsystem name and resubmit the job.

BND2405E **INVALID COMMAND**

Explanation

The command that was entered is not recognized or has invalid syntax.

Module: ADBBMCL

System action:
Processing stops.

User response:
Correct or remove the invalid command and resubmit the job.

BND2406E **SSID MAY ONLY BE SPECIFIED ONCE**

Explanation

Multiple SSID commands were entered. You can issue only one SSID command per execution of Bind Manager.

Module: ADBBMCL

System action:
Processing stops.

User response:
Remove the extra SSID commands and resubmit the job.

BND2407E **MISSING OR INVALID SSID**

Explanation

The SSID command was issued without an operand or with an invalid operand.

Module: ADBBMCL

System action:
Processing stops.

User response:
Correct the SSID command and resubmit the job.

BND2408E **UNABLE TO CONNECT TO SUBSYSTEM *subsystem*, RC=*return code*, REASON=*reason code***

Explanation

Bind Manager is unable to establish a connection to the Db2 subsystem.

subsystem
The subsystem ID.

return code
The return code from Db2.

reason code
The reason code from Db2.

For an explanation of the return code and reason code, see CAF return codes and reason codes (Db2 12 for z/OS).

Module: ADBBMCL

System action:
Processing stops.

User response:
Verify that correct subsystem name was specified the SSID statement and that the subsystem is up and available. Correct the condition that is preventing the connection and resubmit the job.

BND2409S INVALID DATA IN SYSIBM.SYSCOLUMNS, SYMPTOM=*symptom*

Explanation

The Db2 catalog contains invalid data. The catalog might be corrupted.

symptom
The symptom of the failure.

Module: ADBBMCL

System action:
Processing stops.

User response:
Save all output and contact IBM Software Support.

BND2410E UNABLE TO CONNECT TO SUBSYSTEM *subsystem*, BIND REQUIRED

Explanation

Bind Manager is unable to establish a connection to the specified Db2 subsystem with plan ADBBMCL. The plan was not bound in this subsystem, or the bind is out of date.

subsystem
The subsystem ID.

Module: ADBBMCL

System action:
Processing stops.

User response:
Verify that the correct subsystem name was specified in the SSID statement and that the bind for plan ADBBMCL is current. If necessary, bind the plan. Then, resubmit the job.

BND2411I CONNECTED TO SUBSYSTEM *subsystem*

Explanation

Bind Manager successfully connected to the specified Db2 subsystem.

subsystem
The subsystem ID.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

BND2412I DISCONNECTED FROM subsystem *subsystem*

Explanation

Bind Manager disconnected from the specified Db2 subsystem.

subsystem
The subsystem ID.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

BND2414S GETMAIN FAILED IN JFCB ROUTINE

Explanation

Bind Manager could not acquire sufficient working storage.

Module: ADBBMCL

System action:
Processing stops.

User response:
Increase the region size and resubmit the job.

BND2415S DSNTIAR FAILURE, RC=*return code*

Explanation

DSNTIAR issued a non-zero condition code.

return code
The condition code that was issued by DSNTIAR.

Module: ADBBMCL

System action:
Processing stops.

User response:
Verify that the correct version of DSNTIAR is used for the version of Db2 that is running on the specified subsystem. DSNTIAR is accessed based on the STEPLIB DD statement or, if STEPLIB does not include DSNTIAR, the system link list.

BND2416W UNABLE TO PROCESS *dsname* (INVALID RECFM). BYPASSED.

Explanation

The data set is not a load library; it does not have RECFM=U. Therefore, the data set is not processed.

dsname
The name of the data set.

Module: ADBBMCL

System action:
Processing continues.

User response:
Specify a valid load library.

BND2417W UNABLE TO PROCESS *dsname* (NOT PARTITIONED). BYPASSED.

Explanation

The data set is not a partitioned data set (PDS). Therefore, the data set is not processed.

dsname
The name of the data set.

Module: ADBBMCL

System action:
Processing continues.

User response:
Specify a valid load library and resubmit the job.

BND2418W UNABLE TO OPEN *dsname*. BYPASSED.

Explanation

The data set could not be opened for input. Therefore, the data set is not processed.

dsname
The name of the data set.

Module: ADBBMCL

System action:
Processing continues.

User response:
Verify that the data set exists, is not empty, and is not in use by another job or user. Then, resubmit the job.

BND2419E DESERV ERROR, UNABLE TO READ DIRECTORY FOR *dsname*

Explanation

The directory for the data set could not be accessed.

dsname
The name of the data set.

Module: ADBBMCL

System action:
Processing stops.

User response:
Verify that the data set exists, is not empty, and is not in use by another job or user. Then, resubmit the job.

BND2423I NOW PROCESSING *dsname*

Explanation

The load library is being processed.

dsname
The name of the load library.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

BND2425I PACKAGE BYPASSED DUE TO LEVEL PRECOMPILER OPTION Name=*name* Version=*version*

Explanation

The package was precompiled with the LEVEL SQL processing option. However, the ADBBMCL program does not support that option. Therefore, the package is not processed.

This message is issued only once per unique package name or version, regardless of how many collections contain the package.

name
The name of the package.

version
The version ID. This value is displayed only if the package has a version ID.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

Related reference

Descriptions of SQL processing options (Db2 12 for z/OS)

BND2426I *nnnn* PACKAGES WERE BYPASSED DUE TO LEVEL OPTION

Explanation

Some packages in the catalog were not processed, because they were precompiled with the LEVEL SQL processing option. The ADBBMCL program does not support that option.

nnnn
The number of packages in the catalog that were not processed.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

Related reference

Descriptions of SQL processing options (Db2 12 for z/OS)

BND2427I **PACKAGE *package* IS USED BY MODULE *module*, CSECT *csect*; CONTOKEN FOUND AT OFFSET *offset* (HEX) (VERSION=*version*)**

Explanation

A consistency token in a load module matches one from a package in the subsystem. This condition indicates that the package is in use by one or more load modules. Therefore, a FREE command is not generated for this package.

package
The name of the package.

module
The name of the load module.

csect
The name of the control section.

offset
The hexadecimal offset value.

version
The version ID. This value is displayed only if the package has a version ID.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

BND2428E **NO VALUE SPECIFIED FOR COLLID**

Explanation

The COLLID command was entered without an operand.

Module: ADBBMCL

System action:
Processing stops.

User response:
Add a collection name or pattern to the command and resubmit the job.

BND2429E **TOO MANY COLLID COMMANDS**

Explanation

More than 25 COLLID commands were entered. 25 is the maximum number of times that you can issue this command per execution.

Module: ADBBMCL

System action:
Processing stops.

User response:
Remove the extra commands and resubmit the job.

BND2430E **UNSUPPORTED RELEASE OF DB2**

Explanation

ADBBMCL supports DB2 UDB for z/OS 8 and later.

Module: ADBBMCL

System action:
Processing stops.

User response:
Use DB2 UDB for z/OS 8 or later.

BND2432E ***option* was specified without PLIONLY**

Explanation

Either the ALLOWLEVEL or ANYCSECT option was specified without the PLIONLY option.

option
The specified option: ALLOWLEVEL or ANYCSECT.

Module: ADBBMCL

System action:
Processing stops.

User response:
Correct the syntax and resubmit the job.

BND2433E **Invalid option *option* was specified**

Explanation

An invalid option was specified on the OPTIONS statement.

option
The specified option that is not valid.

Module: ADBBMCL

System action:
Processing stops.

User response:
Correct the syntax and resubmit the job.

BND2434E **NO VALUE SPECIFIED FOR PACKAGE.**

Explanation

The PACKAGE command was specified without any package names.

Module: ADBBMCL

System action:
Processing stops.

User response:
Code the command with a package name or range and resubmit the job.

BND2435E TOO MANY PACKAGE COMMANDS

Explanation

More than 25 PACKAGE commands were entered. 25 is the maximum number of times that you can issue this command per execution.

Module: ADBBMCL

System action:
Processing stops.

User response:
Remove the extra commands and resubmit the job.

BND2436E INVALID PACKAGE NAME RANGE.

Explanation

When a name range is specified, the values must be in a sequence such that the lower value is first and the higher value is last. For example, PKG1-PKG5. You submitted a PACKAGE command in which the higher value was specified first.

Module: ADBBMCL

System action:
Processing stops.

User response:
Correct the command and resubmit the job.

BND2438S Unable to open *ddname*

Explanation

Bind Manager was unable to open the LOADLIB data set.

ddname
The DD name that identifies the data set that could not be opened.

Module: ADBBMCL

System action:
Processing stops.

User response:
Verify that the correct DD name is specified.

BND2440W Load module *module-name* has invalid format – bypassed.

Explanation

An ONCODE=8094 condition (S0C4) occurred when processing the load module.

module-name
The name of the load module that caused the error.

Module: ADBBMCL

System action:
Processing for this module is skipped.

User response:
Verify that the module name is correct.

BND2442W MORE THAN 100 DYNAMIC LIBRARIES WERE SPECIFIED - ONLY THIS FIRST 100 WILL BE PROCESSED

Explanation

ADBBMCL can dynamically allocate a maximum of 100 libraries in each execution of the program. Only the first 100 libraries specified are checked.

Module: ADBBMCL

System action:
Processing continues.

User response:
If you want more than 100 libraries to be processed, submit a separate job for each group of 100 libraries.

BND2443W INVALID MODE SPECIFICATION

Explanation

The MODE command was specified with an invalid operand. The only valid operands are DYNAMIC=PROD and DYNAMIC=DEV.

Module: ADBBMCL

System action:
Processing continues.

User response:
Correct the MODE command and resubmit the job.

BND2444W ALLOCATION FAILED, SYMPTOMS=*XXXX XXXX XXXXXXXX*

Explanation

ADBBMCL could not dynamically allocate a library as requested.

XXXX XXXX XXXXXXXX
Diagnostic information in hexadecimal format to be used by IBM Software Support to resolve the problem.

Module: ADBBMCL

System action:
Processing stops.

User response:
Save all output. Contact IBM Software Support.

BND2445W **DEALLOCATION FAILED, SYMPTOMS=*XXXX***

Explanation

ADBBMCL was unable to deallocate a library as requested.

XXXX
Diagnostic information in hexadecimal format to be used by IBM Software Support to resolve the problem.

Module: ADBBMCL

System action:
Processing stops.

User response:
Save all output. Contact IBM Software Support.

BND2446I **NOW PROCESSING DYNAMIC LIBRARY *DSN***

Explanation

The load library is being processed.

DSN
The data set name for the load library.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

BND2447I **PACKAGE *NNN* IN COLLECTION *CCC* IS MARKED "INOPERATIVE"**

Explanation

The package is identified as inoperative in the Db2 catalog.

This message is issued when using dynamic allocation in PROD mode.

NNN
The name of the package.

CCC
The name of the collection.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

BND2448I **PACKAGE *NNN* IN COLLECTION *CCC* IS MARKED "INOPERATIVE" AND WILL BE FREED**

Explanation

The package is identified as inoperative in the Db2 catalog. A FREE command is generated for the package.

This message is issued when using dynamic allocation in DEV mode.

NNN
The name of the package.

CCC
The name of the collection.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

BND2449I **PACKAGE *NNN* IN COLLECTION *CCC* IS MARKED "INVALID"**

Explanation

The package is identified as invalid in the Db2 catalog.

This message is issued when using dynamic allocation in PROD mode.

NNN
The name of the package.

CCC
The name of the collection.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

BND2450I **PACKAGE *NNN* IN COLLECTION *CCC* IS MARKED "INVALID", IS MORE THAN 6 MONTHS OLD, AND WILL BE FREED**

Explanation

The package is identified as invalid in the Db2 catalog. A FREE command is generated for the package.

This message is issued when using dynamic allocation in DEV mode.

NNN
The name of the package.

CCC
The name of the collection.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

BND2451I **UNABLE TO DETERMINE AGE OF INVALID PACKAGE *NNN* IN COLLECTION *CCC***

Explanation

Due to invalid data in the Db2 catalog, Bind Manager is unable to calculate the age of the package. This message is accompanied by message BND2452I.

NNN
The name of the package.

CCC
The name of the collection.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

Related information

"BND2452I" on page 1219
PACKAGE WILL NOT BE FREED

BND2452I **PACKAGE WILL NOT BE FREED**

Explanation

The package that is specified in message BND2451I is not freed.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

Related information

"BND2451I" on page 1219
UNABLE TO DETERMINE AGE OF INVALID PACKAGE *NNN* IN COLLECTION *CCC*

BND2455E **MAXPKG command failed (invalid package count)**

Explanation

The value that is specified for MAXPKG is not a number.

Module: ADBBMCL

System action:
Processing stops.

User response:
Specify a number for MAXPKG.

BND2456E **MAXLM command failed (invalid load module count)**

Explanation

The value that is specified for MAXLM is not a number.

Module: ADBBMCL

System action:
Processing stops.

User response:
Specify a number for MAXLM.

BND2457I **Package limit was reached; the entry listed below will be the last one checked – Package *package-id* – Version *version-id* – In Collection *collection-id***

Explanation

The MAXPKG value was reached.

package-id
The package ID of the last package that is to be checked.

version-id
The version ID of the last package that is to be checked.

collection-id
The collection ID for the last package that is to be checked.

Module: ADBBMCL

System action:
No packages after the listed package are checked.

User response:
Increase the MAXPKG value if needed.

BND2458I **Load module limit was reached – Last module checked was *module* in *dynamic-dsn***

Explanation

The MAXLM value was reached.

module
The module name.

dynamic-dsn
The name of the dynamically allocated data set.

Module: ADBBMCL

System action:
No modules after the listed module are checked.

User response:
Increase the MAXLM value if needed.

BND2459W DBRM generation failed for package *package-id* – In collection *collection-id* – ADBGEN return code= *return-code*

Explanation

A backup copy of the DBRM was not generated for the package.

package-id
The package ID.

collection-id
The collection ID for the package.

return-code
The return code from ADBGEN.

Module: ADBBMCL

System action:
Processing continues.

User response:
Check the validity of the package.

BND2460W DBRM generation failed for package *package-id* – In collection *collection-id* – ADBBMGN return code= *return-code*

Explanation

A backup copy of the DBRM was not generated for the package.

package-id
The package ID.

collection-id
The collection ID for the package.

return-code
The return code from ADBBMGN.

Module: ADBBMCL

System action:
Processing continues.

User response:
Check the validity of the package.

BND2462S Checkpoint file is empty. Restart is not possible.

Explanation

The checkpoint is terminated.

Module: ADBBMCL

System action:
Processing stops.

User response:
Rerun the job from the beginning.

BND2463S Invalid checkpoint file, symptom=*symptom*. Restart failed.

Explanation

The checkpoint is terminated, because the checkpoint record type is invalid.

symptom
The checkpoint record type.

Module: ADBBMCL

System action:
Processing stops.

User response:
Rerun the job from the beginning.

BND2464E Run halted before first checkpoint; no restart is possible. Rerun job from the beginning.

Explanation

The checkpoint phase is 1. The first checkpoint was not taken, and the checkpoint is terminated.

Module: ADBBMCL

System action:
Processing stops.

User response:
Rerun the job from the beginning.

BND2468W Module *modname* bypassed, unsupported RMODE. See message documentation for more information.

Explanation

Linkage Editor attributes RMODE(SPLIT) and RMODE(64) are not currently supported.

modname
The name of the bypassed load module.

Module: ADBBMCL

System action:
Processing continues.

User response

No action is required.

BND2469W **Generating DBRM bind information failed for package *package-id* in collection *collection-id* *procedure-name* return code= *return-code* Message: *message-text***

Explanation

A backup copy of the DBRM was not generated for the package.

package-id
The package ID.

collection-id
The collection ID for the package.

procedure-name
The name of the procedure that detected an error.

return-code
The return code from the *procedure-name* procedure.

message-text
The error explanation from the *procedure-name* procedure.

Module: ADBBMCL

System action:
Processing continues.

User response:
Check the message from *procedure-name* and correct the errors.

BND2498S **Internal error in ADBBMCL – processing terminated**

Explanation

An unexpected error occurred in the Bind Manager clean-up feature.

Module: ADBBMCL

System action:
Processing stops.

User response:
Save all output. Contact IBM Software Support.

BND2706S **A runtime error occurred in *xxxxxxxx*, symptom=*yyyyyyyy*.**

Explanation

An internal processing error in program logic occurred.

xxxxxxxx
A diagnostic code for use by IBM Software Support.

yyyyyyyy
A diagnostic code for use by IBM Software Support.

Module: ADBBMRG

System action:
Processing stops.

User response:
Save all output and contact IBM Software Support.

BND2909I **MEMBER *member_name* is not found in DDNAME DBRMIN**

Explanation

The DBRM name that was specified in the BIND command was not found in the library that is identified by the DBRMIN DD statement. The BIND command is passed to BINDOUT. However, if the same library is used, processing of the BINDOUT DD statement will also fail.

member_name
The member name that was specified for the DBRM.

Module: ADBBMA3

System action:
Processing continues.

User response:
Correct the DBRM name or the DBRMIN DD statement.

BND2911E **INPUT PARAMETER *pname* NOT RECOGNIZED**

Explanation

The bind avoidance program does not recognize the parameter *pname*. The parameter *pname* is ignored.

pname
The incorrect parameter text.

Module: ADBBMA3

System action:
Processing continues. The return code is set to 8.

User response:
Correct the parameter and run the ADBBMA3 program again.

BND2917I **SYSPACKAGE ENTRY=*dbrm_name* WITH VERSION *version* NOT FOUND**

Explanation

The new DBRM specified a version that does not exist in SYSIBM.SYSPACKAGE. The BIND command must be processed.

dbrm_name
The name of the DBRM.

version
The specified version identifier for the package.

Module: ADBBMA3

System action:
Processing continues.

User response

No action is required.

BND2918I SYSPACKAGE ENTRY=*dbrm_name* WITH VERSION *version* HAD CONTOKEN *contoken_value* AND MATCHES NEW DBRM

Explanation

The new DBRM specified a version that exists in SYSIBM.SYSPACKAGE, and the consistency token matches. The BIND command is not processed.

dbrm_name
The name of the DBRM.

version
The specified version identifier for the package.

contoken_value
The specified consistency token for the package.

Module: ADBBMA3

System action:
Processing continues.

User response

No action is required.

BND2919I SYSPACKAGE ENTRY=*dbrm_name* WITH VERSION *version* HAD CONTOKEN *contoken_value* AND DOES NOT MATCH NEW DBRM

Explanation

The new DBRM specified a version that exists in SYSIBM.SYSPACKAGE. However, the consistency token (CONTOKEN) does not match. A BIND command is generated unless the ADBBMA3 option UNIQUE-VERSION was specified, in which case message BDN2920E is issued instead.

dbrm_name
The name of the DBRM.

version
The specified version identifier for the package.

contoken_value
The specified consistency token for the package.

Module: ADBBMA3

System action:
Processing continues.

User response

No action is required.

Related tasks

"Determining whether applications need to be rebound" on page 888

Related information

"BND2920E" on page 1222
VERSION *version* IS NOT UNIQUE FOR PACKAGE *package-name* IN COLLECTION *collid*. SETTING RC=8 BECAUSE OPTION UNIQUE-VERSION WAS SPECIFIED

BND2920E VERSION *version* IS NOT UNIQUE FOR PACKAGE *package-name* IN COLLECTION *collid*. SETTING RC=8 BECAUSE OPTION UNIQUE-VERSION WAS SPECIFIED

Explanation

Although the package consistency token (CONTOKEN) changed, the package version is the same in both the Db2 catalog and the DBRM. Because the ADBBMA3 option UNIQUE-VERSION was specified, the BIND command is not generated.

version
The specified version identifier for the package.

package-name
The name of the package.

collid
The collection ID.

Module: ADBBMA3

System action:
ADBBMA3 continues scanning all BIND commands in the BINDIN data set and then ends with return code 8.

User response:
If you do not want this package bound, no action is necessary. Otherwise, run ADBBMA3 again without UNIQUE-VERSION.

Related tasks

"Determining whether applications need to be rebound" on page 888

BND2935E **There is a DBRMLIB process failure. The DDNAME *ddname* is missing.**

Explanation

ADBBMA3 requires the BINDIN, BINDOUT, and DBRMIN DD statements. Because one of these statements is missing, ADBBMA3 cannot continue processing.

The SYSPRINT DD statement is not required, but only SYSPRINT identifies the location for any error messages.

ddname
The DD name for the missing statement.

Module: ADBBMA3

System action:
Processing stops. The return code is set to 16.

User response:
Provide the missing DD statement and run the ADBBMA3 program again.

BND2940I **The BINDIN control cards do not specify EXPLAIN(YES).**

Explanation

Access path analysis is not performed for this bind operation, because the BIND command does not specify EXPLAIN or specifies EXPLAIN(NO).

Module: ADBBMA3

System action:
Processing continues.

User response:
If you want access path analysis to be done, specify EXPLAIN(YES) on the BIND command.

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.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offerings use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at http://www.ibm.com/privacy/details and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Index

Product Number: 5655-DT2

SC27-8898-04