CA Database Management Solutions for DB2 for z/OS

Value Pack Reference Guide

Version 17.0.00, Second Edition

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2010 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Database Analyzer™ for DB2 for z/OS (CA Database Analyzer)
- CA Fast Index® for DB2 for z/OS (CA Fast Index)
- CA Fast Load for DB2 for z/OS (CA Fast Load)
- CA Fast Recover™ for DB2 for z/OS (CA Fast Recover)
- CA Fast Unload® for DB2 for z/OS (CA Fast Unload)
- CA Merge/Modify™ for DB2 for z/OS (CA Merge/Modify)
- CA Plan Analyzer® for DB2 for z/OS (CA Plan Analyzer)
- CA Quick Copy for DB2 for z/OS (CA Quick Copy)
- CA Rapid Reorg® for DB2 for z/OS (CA Rapid Reorg)
- CA RC/Migrator™ for DB2 for z/OS (CA RC/Migrator)
- CA RC/Query® for DB2 for z/OS (CA RC/Query)
- CA RC/Secure™ for DB2 for z/OS (CA RC/Secure)
- CA RC/Update™ for DB2 for z/OS (CA RC/Update)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made since the last release of this documentation:

- ACM Upgrade Considerations (see page 63)—Updated the description of this topic.

Contents

Chapter 1: Introduction

This guide describes the Value Pack components that are provided at no additional charge with the CA Database Management Solutions for DB2 for z/OS.

This section contains the following topics:

Components

The following Value Pack components are automatically installed with the CA Database Management Solutions for DB2 for z/OS:

- Alternate catalog mapping facility to create, update, copy, and delete alternate catalog maps
- Batch processor to execute jobs (online or in batch mode)

  Note: For more information about the Batch Processor interface, see the *CA Database Management Solutions for DB2 for z/OS Batch Processor Reference Guide*.
- DB2 command processor to execute CA and DB2 commands and request syntax help
- Interactive SQL (ISQL) to execute SQL statements contained within a data set
- Thread termination and dynamic DSNZPARM to terminate DB2 threads, perform dynamic DSNZPARM alteration updates, and automatically set DSNZPARMS without recycling DB2

  Note: For more information about terminating threads and updating DSNZPARM, see the *CA Database Management Solutions for DB2 for z/OS Thread Termination\Dynamic DSNZPARM User Guide*.
- Utility manager to create, view, update, and submit utility extracts
- Referential integrity management (RI Manager) to track, organize, and manage DB2 system and application enforced RI relationships.

  Note: For more information about the RI Manager, see the *CA Database Management Solutions for DB2 for z/OS RI Manager User Guide*.

Authorization and Security

You must have appropriate authorization and security to execute the Value Pack components. Many of the security restrictions are implemented by the DB2 security system. DB2 security is never bypassed.

The Product Authorizations facility lets you control user privileges for products, tables, and DB2 system catalog tables. This facility provides an online method of executing the GRANT and REVOKE statements you need to control security.

Note: For more information about the Product Authorizations facility, see the *CA Database Management Solutions for DB2 for z/OS General Facilities Reference Guide*.

General Function Authorizations

You can grant and revoke access through the Product Authorizations facility to the following Value Pack component plans:

General Functions

Lists the plans for the Value Pack components and general facilities. Select from the following plans:

Alt. Catalog Mapping

Provides authorization to execute the Alternate Catalog Mapping facility.

DB2 Command Processor

Provides authorization to execute the DB2 command processor.

Batch Processor

Provides authorization to execute the Batch Processor interface.

Note: The Batch Processor interface is used by many products. If you do not have authority to execute this interface, you will not be able to access those functions in the products.

Interactive SQL

Provides authorization to execute Interactive SQL.

Note: Authorization for Batch Processor is also required to use the Interactive SQL facility.

RI Manager

Provides read-only or full command (execute) access to the applicable RI manager functions. Select from the following plans:

RI Manager Services

Provides full command access to RI Manager. Full command access lets you view the RI relationships in RI structures and use all line commands to create, modify, and delete RI relationships.

RI Manager - R/O

Provides read-only access to RI Manager. Read-only access lets you view the RI relationships of complicated RI structures while preserving the integrity of the RI relationships. You cannot execute any line commands.

In addition, the following authorizations are needed to the RI Manager tables:

- SELECT authority for read-only access
- SELECT, INSERT, UPDATE, and DELETE authorities for full access
- SELECT, INSERT, and DELETE authorities to add user-child relationships

Thread Term/Dynam DSNZ

Lists the plans for functions within Thread Termination and dynamic DSNZPARM. Select from the following plans:

Required Install Step

Regardless of whether you use the external security system to handle authorizations it is necessary to use Product Authorizations to grant execute authority for the Manage DB2 Default Group and View Subsystem Data plans to the USERID associated with the DB2 subsystem DBM1 address space.

Browse Monitor Profile

Browse but not update the thread monitor profiles.

View Subsystem Data

View DB2 subsystem data including DB2 thread information and DB2 parameters.

Terminate Threads

Terminate threads in online and batch modes.

Update Monitor Profile

Update the thread monitor profiles.

Manage Scheduler

Start, stop, and refresh the scheduler and maintain a list of schedules.

Manage DB2 Default Group

Update (manage) the DB2 subsystem. This is the default security group. Unless the management commands and functions are defined to a specific user security profile, they will be controlled by this default security profile.

Manage DB2 User Grp1 through Grp5

Assign access to specific management commands and functions to user security groups to limit access of these functions.

PTT View Host Variab

Includes input host variables associated with an SQL statement on the Thread Termination Active Threads SQL Call Text Display.

Grant Product Authorizations

You can grant or revoke authority through the Product Authorization facility to the Value Pack components.

Note: All users require EXECUTE authority on the product plan for all DB2 subsystems on which you want to use the product. This means you must grant access to the user on all DB2 subsystems on which the product or component is used.

To grant product authorizations

1. Type **A** on the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The Product Authorizations panel appears.

2. Type **E** next to the component to see a complete list of related plans and press Enter.

   The list of related plans appears.

3. Type **G** in the line command next to the function, the user's ID in the TO ID field and press Enter.

   Authorization is granted to the function.

   Note: If the grant is successful, a G or GW appears in the TO column and Authority Granted appears in the MESSAGES column.

Chapter 2: Using the Batch Processor

To execute jobs, you can use the batch processor in online or batch mode.

You can specify the following options for the batch processor:

- Execution Specifications. These options control the processing of input data sets.
- Batch Processor Input Data Set. These options identify which data set the batch processor will process.
- Processing options. The processing options let you control various execution choices.
- Audit options. The audit options let you specify how the batch processor handles the generated output.

Note: For more information about using the Batch Processor, see the *Batch Processor Reference Guide*.

Chapter 3: Using the DB2 Command Processor

This section contains the following topics:

DB2 Command Processor

The DB2 Command Processor executes DB2 commands from within a CA Database Management Solutions for DB2 for z/OS product session and provides help with command syntax. The DB2 Command Processor lets you perform the following tasks:

- Request syntax help
- Execute CA and DB2 commands
- View a list of executed commands

Request Command Syntax Help

You can receive help on the syntax of any supported DB2 command using the DB2 Command Processor.

Note: For details about command syntax, see the IBM *DB2 Command and Utility Reference Guide*.

To request command syntax help

1. Type **C** on the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The DB2 Command Processor panel appears. The ID of the DB2 subsystem that you are currently connected to appears on this panel and the valid START, STOP, DISPLAY, and other commands. You change the DB2 subsystem if needed.

   Note: You can also access the DB2 command processor using the DB2C <command> Interproduct Interface command.

2. Type one of the following two letter command codes in the Enter Utility Code for Syntax Help field:
   - The following DISPLAY commands are supported:
     - DD (database)
     - DR (rlimit)
     - DX (trace)
     - DT (thread)
     - DL (location)
     - DA (archive)
     - DB (bufferpool)
     - DU (utility)
     - DC (CA utility for DB2 for z/OS)
     - DG (group)
     - DO (group bufferpool)
     - DM (procedure)
     - DF (DDF)
     - DS (function specific)
     - DH (CA history)
     - DZ (CA active)
   - The following STOP commands are supported:
     - XD (database)
     - XR (rlimit)
     - XX (trace)
     - XF (DDF)
     - XB (DB2)
     - XP (procedure)
   - The following START commands are supported:
     - SD (database)
     - SR (rlimit)
     - SX (trace)
     - SF (DDF)

- SG (procedure)
- SS (function specific)

- The following other commands are supported:
  - AB (ALTER BUFFERPOOL)
  - AC (ALTER CA UTILID)
  - AG (ALTER GROUPBUFFERPOOL)
  - AU (ALTER UTILITY)
  - AR (ARCHIVE LOG)
  - CF (CANCEL DDF THREAD)
  - CT (CANCEL THREAD)
  - MX (MODIFY TRACE)
  - RB (RECOVER BSDS)
  - RE (RESET INDOUBT)
  - RG (RESET GENERICLU)
  - RI (RECOVER INDOUBT)
  - RP (RECOVER PSTPONED)
  - SA (SET ARCHIVE)
  - TC (TERM CA UTILITY)
  - TU (TERM UTILITY)

  Press Enter.

  The syntax help panel for the requested command appears and provides the following information:

  - The beginning syntax for the command appears in the DB2 Command field.
  - The required and optional parameters for the command follow the beginning syntax information. Required parameters are highlighted. Stacked parameters only require one entry. For example, you can specify only one of the following parameters with the -MODIFY TRACE command: GLOBAL, ACCTG, STAT, AUDIT, and MONITOR.

3. (Optional) Press Enter to execute the command.

   The results appear on the DB2 Command Processor panel. Related messages about the execution of the command appear at the bottom of the data.

Execute DB2 and Special Commands

The DB2 Command Processor lets you execute DB2 and special commands (-ALTER CA, -DISPLAY CA, or -TERM CA).

To execute DB2 and special commands

1. Type **C** on the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The DB2 Command Processor panel appears.

2. Type a DB2 or special command in the area under Enter a DB2 Command and press Enter.

   The command is executed and the results appear on the panel. This includes any syntax errors.

View a List of Commands

You can request a list of the last 15 commands entered and then request syntax help, execute, or delete the commands from the list using the DB2 Command Processor.

To view a list of commands

1. Type **C** on the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The DB2 Command Processor panel appears.

2. Type a question mark (?) in the area under the Enter a DB2 Command field and press Enter.

   A list of the last 15 commands that were entered appears listed from most recent to oldest by global, product, and panel specific.

3. (Optional) Type one of the following commands next to a command in the list:

   S

   Request syntax help for the command.

   E

   Execute the command.

   D

   Delete the command from the selection list.

   Press Enter.

   The command is processed as requested.

CA Commands

You can use the -ALTER CA, -DISPLAY CA, and -TERM CA commands in the DB2 Command Processor to perform the following actions:

- Redefine the DEFER keyword value in CA Rapid Reorg jobs
- View a list of the DB2 utilities that are in the DB2 utility restart table
- Terminate one or more CA utilities

-ALTER CA Command—Override the DEFER Keyword Value

The -ALTER CA command redefines the defer value that has been set by the DEFER keyword in a CA Rapid Reorg job. The defer value indicates the number of minutes to stop the log apply phase of a high performance reorg.

If you have set the DEFER value high in your CA Rapid Reorg job (or to 99, which will stop the job indefinitely), you can manually change it to a lower value or set it to zero using the -ALTER CA command.

Note: For more information about defer processing, see the *CA Rapid Reorg for DB2 for z/OS User Guide*.

This command has the following format:

```
-ALTER CA UTILID(utilid) DEFER(int)
```

UTILID(*util-id*)

Restricts the command to only those utilities with the specified utility ID. You can specify a wild card character for this value.

DEFER(*int*)

Specifies the number of minutes from 0 to 99 to stop the log apply phase of an online mode reorg,

Default: 0

-DISPLAY CA Command—Display CA Utility Information

The -DISPLAY CA command provides the following type of information about the CA Database Management for DB2 for z/OS utilities:

- All utilities operating against an object in a database
- All utilities operating against a partition
- All utilities with a specific utility ID and optional checksum specification

- Active utility information (like PHASE and ROWCOUNT)
- Active utility types
- All utilities submitted by a specific user ID or mask
- Summary information

Note: To execute the -DISPLAY CA command, you must have the following authorities: SELECT authority on the PTI.PTGLxxx_RESTART and PTI.PTGLxxx_RESTART2 table or SYSADM authority. (Note that xxx can vary with each release.) You must also have EXECUTE authority on the PUT General Utilities plan. For more information about providing authorization to the utilities plan, see the *CA Database Management Solutions for DB2 for z/OS General Facilities Reference Guide*.

This command has the following format:

```
-DISPLAY CA or -DIS CA
```

More information:

CA Command Syntax (see page 23)

-TERM CA Command—Terminate CA Utilities

The -TERM CA command terminates one or more CA Database Management utilities for DB2 for z/OS. This command only deletes entries from the restart table that are no longer applicable. It does not terminate a utility job that is currently executing. It also does not reset the status of objects that the utility was operating against to the status prior to the initiation of the utility. To terminate an active utility, cancel the job. To restore objects of a cancelled utility execution to the status prior to the utility execution, submit the cancelled job with the RESTART(TERM) parameter. See the user guide for the utility for more information.

Note: To execute the -TERM CA command, you must have DELETE authority on the PTI.PTGL*xxx*_RESTART and PTI.PTGL*xxx*_RESTART2 table or SYSADM authority. (Note that *xxx* can vary with each release.)

This command has the following syntax:

```
-TERM CA
```

More information:

CA Command Syntax (see page 23)

CA Command Syntax

You can use the following syntax with the -DISPLAY CA and -TERM CA commands:

```
[DATABASE{(database-name)}|{(database-name)}]
[SPACENAM{(spacename)}|OBJNAME{(spacename)}]
[PART {(partition-number)}]
[UTILITY{(utility-id)}|UTILID{(utility-id)}|UTIL{(utility-id)}]
[CHECKSUM {(checksum-value)}]
[UTYPE {(utility-type)}]
[USERID {(userid)}]
[SUMMARY]
[ACTIVE]
```

The following describes these syntax values:

DATABASE (*database-name*) or (*database-name*)

(Optional) Restricts the -DISPLAY CA or -TERM CA commands to only those utilities that are operating against a specific database.

- The DATABASE keyword is optional in a database specification. If you include it, you must follow it with a database name enclosed in parentheses.
- You can also specify only the database name enclosed in parentheses without the DATABASE keyword. A database specification that does not include the DATABASE keyword does not need to immediately follow the CA keyword. Any value enclosed in parentheses in the command that is not preceeded by a keyword is treated as a database specification.
- Wild card characters can be used.

SPACENAM(*spacename*) or OBJNAME(*spacename*)

(Optional) Restricts the -DISPLAY CA or -TERM CA commands to only those utilities that are operating against a specific space name. Wild card characters can be used.

Note: The following are also accepted forms for these keywords: SPACE, SPACEN, SPACENA, SPACENAM, SPACENAME, OBJ, OBJN, OBJNA, OBJNAM, and OBJNAME.

PART (*partition-number*)

(Optional) Restricts the -DISPLAY CA or -TERM CA commands to only those utilities that are operating against a specific partition of a tablespace or indexspace.

Note the following:

- You can specify a partition number of 0; however, it does not impact processing.
- PART only impacts the utilities that are selected to be viewed or terminated. It does not impact the partitions that appear in the output of the -DISPLAY CA command.

- PART(1) does not select utilities that are operating against a non-partitioned object.

UTILITY (*utility-id*), UTILID (*utility-id*), or UTIL (*utility-id*)

(Optional) Restricts the -DISPLAY CA or -TERM CA commands to only those utilities with the specified utility ID. Wild card characters can be used.

Important! It is possible to have more than one utility with the same utility ID. If this occurs and you want to restrict a -DISPLAY CA or -TERM CA command to only one of these utilities, use additional keywords (for example, CHECKSUM or UTYPE).

CHECKSUM (*checksum-value*)

(Optional) Restricts the -DISPLAY CA or -TERM CA commands to only those utilities with a specific checksum value. This keyword is useful if there are multiple utilities with the same utility ID. You can determine the checksum value from the output of the -DISPLAY CA command.

UTYPE(*utility-type*)

(Optional) Restricts the -DISPLAY CA or -TERM CA commands to utilities of the specified type. You can specify any of the following values (that correspond to the UTYPE that appears in the output of the -DISPLAY CA command):

Note: You can also specify a global wild card like UTYPE(*); however, it has the same affect as not specifying the keyword.

BUILD

Indicates the utility type is CA Fast Index.

Valid synonyms: FASTBLD and FASTINDEX

CHECK

Indicates the utility type is CA Fast Check.

Valid synonym: FASTCHECK

COPY

Indicates the utility type is CA Quick Copy.

LOAD

Indicates the utility type is CA Fast Load.

Valid synonym: FASTLOAD

PDA

Indicates the utility type is CA Database Analyzer.

Valid synonym: PDASTATS

PMM

Indicates the utility type is CA Merge/Modify.

Valid synonyms: MODIFY, ACCUMULATE, and MERGECOPY

PPX

Indicates the utility type is CA Partition Expert.

Valid synonyms: PPXROLL, PPXANALYSIS

REBUILD

Indicates the utility type is CA Fast Recover using REBUILD.

RECOVER

Indicates the utility type is CA Fast Recover.

Valid synonyms: REBUILD, RECVR-TS, and RECVR-IX

REORG

Indicates the utility type is CA Rapid Reorg.

Valid synonyms: REORG-TS and REORG-IX

UNLOAD

Indicates the utility type is CA Fast Unload.

Valid synonym: FASTUNLOAD

USERID (*user-id*)

(Optional) Restricts the -DISPLAY CA or -TERM CA commands to those utilities submitted by the specified user ID. Wild card characters can be used.

SUMMARY

(Optional) Excludes space name information from the output of the -DISPLAY CA command.

Note: If specified with -TERM CA command, it has no impact.

ACTIVE

(Optional) Shows the utility status and unload phase and count when used with -DISPLAY CA UTILID(*utility-id*).

Specify RESTART-TABLE YES in the PFU parmlib or AUTO-RESTART YES in the UTIL parmlib to show active utility information for rows unloaded.

Naming Conventions

To specify a database name, spacename, utility-id, or user ID, use *one* of the following:

- A specific name
- One or more letters followed by an asterisk (*). For example, BR13* selects BR132SP.

- A series of letters and asterisks (*). For example, BR*3*S* selects BREK3QSP and BR132SP, but not PBRA3ASP or B2RE3LS.
- A single asterisk (*) to select all objects

Parsing Rules

Use parentheses to mark the end of a word and the beginning of a new word. A blank space marks the end of the current word and the next non-blank marks the beginning of the next word. This means blanks are not required before and after a value enclosed in parentheses. For example, the following three commands are equivalent:

```
-DISPLAY CA UTILID (USER1.TSEXAMP) CHECKSUM (62)
-DISPLAY CA UTILID(USER1.TSEXAMP) CHECKSUM(62)
-DISPLAY CA UTILID(USER1.TSEXAMP)CHECKSUM(62)
```

Examples

Examples are provided for using the -DISPLAY CA and -TERM CA commands.

-DISPLAY CA Command Examples

Use the following examples to execute the -DISPLAY CA command on your DB2 subsystems and list CA Database Management Solutions for DB2 for z/OS utilities.

Example: List All Utilities

To list *all* utilities, use any of the following command syntax:

```
-DISPLAY CA
-DISPLAY CA UTILITY (*)
-DISPLAY CA UTIL (*)
-DISPLAY CA UTILID(*)
```

The following commands also list all utilities because the database name is a global wildcard:

```
-DISPLAY CA (*)
-DISPLAY CA DATABASE(*)
```

Example: List All Utilities for a Database

Use *either* of the following commands for displaying the utilities that are operating against objects in database TESTDB:

```
-DISPLAY CA (TESTDB)
-DISPLAY CA DATABASE(TESTDB)
```

Use *either* of the following commands for displaying the utilities that are operating against objects in any database that begins with AB:

```
-DISPLAY CA (AB*)
-DISPLAY CA DATABASE(AB*)
```

Example: List All Utilities for a Database and Spacename Specification

Use the following command for displaying all utilities operating against an object in database TESTDB with a spacename of TESTTS:

```
-DISPLAY CA (TESTDB) SPACENAM(TESTTS)
```

Example: List All Utilities for a Database and Spacename Mask

Use the following command for displaying all objects in database TESTDB with space name that begins with TEST:

```
-DISPLAY CA (TESTDB) SPACENAM(TEST*)
```

Example: List All Utilities for a Specific Partition

Use the following command for displaying all utilities operating against partition 3 of an object in database TESTDB with a space name of TESTTS:

```
-DISPLAY CA (TESTDB) SPACENAM(TESTTS) PART(3)
```

Example: List All Utilities for a Specific Utility ID

Use the following command for displaying all utilities with a utility-id that begins with AB:

```
-DISPLAY CA UTILITY(AB*)
```

Use the following command for displaying all utilities with a utility-id of USER1.TSEXAMP:

```
-DISPLAY UTILID(USER1.TSEXAMP)
```

Example: List a Utility with a Specific Utility-ID and Checksum

To list only the utility with a utility-id of USER1.TSEXAMP whose checksum value is 62, specify the following:

```
-DISPLAY CA UTILID(USER1.TSEXAMP) CHECKSUM(62)
```

Example: List Active CA Fast Load and CA Rapid Reorg Information

To list active utility information (like PHASE and ROWCOUNT) for CA Fast Load and CA Rapid Reorg, specify the following:

```
-DISPLAY CA UTILID(utility-id) ACTIVE
```

Example: List All CA Fast Load Utilities

To list all CA Fast Load utilities, specify the following:

```
-DISPLAY CA UTYPE(LOAD)
```

Example: List All Utilities Submitted by a Specific User ID

To list all utilities submitted by user ID USER1, specify the following:

```
-DISPLAY CA USERID(USER1)
```

To list all utilities submitted by any user IDs that begin with AB, specify the following:

```
-DISPLAY CA USERID(AB*)
```

Example: List Summary Information

To list summary information for all utilities, specify the following:

```
-DISPLAY CA SUMMARY
```

-DISPLAY CA Sample Execution Results

The following sample execution results show the output of the -DISPLAY CA command:

```
-DISPLAY CA (POR154)

PUT3003I  UTILID        = USERABC.EMPLOYEE     CHECKSUM = 10802
          UTYPE         = REORG                USERID   = USERABC
          TIME ACCESSED = 2004-01-14 13.07.51

          DATABASE: POR154     OBJECT: POR154    PART: 1
          DATABASE: POR154     OBJECT: POR154    PART: 2
          DATABASE: POR154     OBJECT: POR154X1  PART: 1
          DATABASE: POR154     OBJECT: POR154X1  PART: 2
          DATABASE: POR154     OBJECT: POR154X2

PUT3004I  Number of CA-DB2 utilities displayed: 1
```

The following sample execution results show the output of the -DISPLAY CA command with the SUMMARY keyword:

Note: The space name information does not appear in the output when SUMMARY is specified.

```
-DISPLAY CA (POR154) SUMMARY

PUT3003I  UTILID        = USERABC.EMPLOYEE     CHECKSUM = 10802
          UTYPE         = REORG                USERID   = USERABC
          TIME ACCESSED = 2004-01-14 13.07.51

PUT3004I  Number of CA-DB2 utilities displayed: 1
```

-TERM CA Command Examples

Use the following examples to execute the -TERM CA command on your subsystems.

Example: Terminate All CA Utilities for a Database

To terminate all CA utilities that are operating against objects in database TESTDB, specify the following:

```
-TERM CA (TESTDB) or -TERM CA DATABASE(TESTDB)
```

Example: Terminate All CA Utilities for a Database and Spacename Specification

To terminate all CA utilities for DB2 for z/OS operating against an object in database TESTDB with a spacename of TESTTS, specify the following:

```
-TERM CA (TESTDB) SPACENAM(TESTTS)
```

Example: Terminate All CA Utilities for a Specific Partition

To terminate all CA utilities for DB2 for z/OS operating against partition 12 of an object in database TESTDB with a space name of TESTTS, specify the following:

```
-TERM CA (TESTDB) SPACENAM(TESTTS) PART(12)
```

Example: Terminate All CA Utilities with a Specific Utility ID

To terminate all CA utilities for DB2 for z/OS with a utility-id of USER1.TSEXAMP, specify the following:

```
-TERM UTILID(USER1.TSEXAMP)
```

Example: Terminate Utility-ID and Checksum Specification

To terminate only the utility with a utility-id of USER1.TSEXAMP whose checksum value is 62, specify the following:

```
-TERM CA UTILID(USER1.TSEXAMP) CHECKSUM(62)
```

Example: Terminate All CA Fast Load Utilities Running Against a Specific Database

To terminate all CA Fast Load utilities that were operating against objects in the database named TESTDB, specify the following:

```
-TERM CA DATABASE(TESTDB) UTYPE(LOAD)
```

Example: Terminate All Utilities Submitted by a Specific User

To terminate all CA utilities for DB2 for z/OS submitted by user ID USER1, specify the following:

```
-TERM CA USERID(USER1)
```

How To Execute Commands in Batch Mode

The DB2 Command Processor lets you execute any DB2C command in batch mode using JCL.

Note: Sample JCL is provided in the *high-level*.CDBASRC member DB2CJCL.

To execute commands in batch mode, enter commands in an unnumbered data set for SYSIN. The DB2 Command Processor interprets a character in column 80 as a continuation indicator. If the command input data set contains line numbers in columns 73-80, you will not receive SYSPRINT output.

Consider the following when you create this data set:

- Begin the commands with a hyphen in column one followed by the command. For CA commands, there must only be one space between the command and CA keyword (-DISPLAY CA, -TERM CA).
- The PARM section of your EXEC statement to include your DB2 subsystem ID and the following program information:

  EP=PTLDB2C/*DB2-subsystem-ID*
- The input must come from the SYSIN DD. This can be in-stream or from a data set.

Save your changes and submit the JCL for execution.

The output is directed to the output device specified by the SYSPRINT DD statement. Most of the output is contained in an 80-column report, similar to the online output. However, the batch output is 132 columns wide. Columns 81 through 132 only contain an input command indicator to identify the first statement of each DB2C command input statement.

How To Execute Commands from the Console

You can execute any DB2C command from the console (one at a time).

To execute commands from the console, follow these steps:

1. Modify the RUNDB2C JCL in *high-level*.CDBASRC as follows before executing DB2C commands from the console for the first time:

   a. In the PROC statement, enter the default SSID. Whenever you execute the procedure, you will not need to include SSID information if you are executing it on the default SSID.

   b. For STEPLIB, enter your CA load library and DB2 load library.

   c. For PTILIB, enter your CA load library and DB2 load library.

   d. For PTIPARM, enter your CA parameter library.

   Save your changes.

   The member is saved. The values you specified are used as defaults when the DB2C command is executed.

2. Copy the edited version to your system procedure library.

   The default SSID and CA and DB2 load libraries are set for use in the command syntax.

3. Use the following syntax to execute any DB2 command from the console (one at a time):

   ```
   S RUNDB2C,SSID=ssid,CMD='db2c command'
   ```

   S RUNDB2

   Starts the RUNDB2 procedure.

   SSID=*ssid*

   Identifies a valid subsystem ID, unless you want to use your default SSID as defined in the RUNDB2C JCL..

   CMD='*db2c command*'

   Identifies the DB2C command in single quotes.

   Press Enter.

   The command is executed.

Example: Execute -DIS CA on D91A

To run the -DIS CA command on the D91A subsystem, specify the following:

```
S RUNDB2C,SSID=D91A,CMD='-DIS CA(*)'
```

Chapter 4: Using Interactive SQL

This section contains the following topics:

Interactive SQL

DB2 does not provide a facility for testing SQL entered into a program during the development stage. The application programmer must precompile the program to find any syntax errors and then execute the program to discover if the SQL statements perform the desired operations. This is a very time-consuming, costly, and tedious process.

The Interactive SQL (ISQL) facility lets you execute SQL statements contained in a data set. This provides a faster method of testing SQL statements without precompiling and executing the entire program.

You can use ISQL to perform the following tasks:

- Execute statements in embedded COBOL, PL/I, or Assembler code.
- Execute multiple SELECT statements. The results of each SELECT statement are returned before continuing. Users can browse and print the returned rows before continuing execution.
- Enter a Row Limit for SELECT statements to reduce the number of rows returned.
- Specify Host Variable Help. Before execution of SQL, ISQL will parse out the host variables.
- Execute extracted SQL statements in batch or online mode.
- Enter and execute SQL statements using the interactive SQL facility.
- Use Row Indicator for Selects. The SQL Editor returns the results of a query, and the number of rows returned.

Start ISQL

You can start ISQL from the CA Database Management Solutions for DB2 for z/OS Main Menu, from any CA Database Management Solutions for DB2 for z/OS product command line, from the TSO command line outside of the products, from an ISPF edit session, or from the TSO Ready prompt as described in the following sections.

Note: ISQL uses the ISQL Online SQL Execution and SQL Editor screens to execute SQL statements. See ISQL Online SQL Execution and SQL Editor for detailed information about the available options on these screens.

More information:

ISPF Online SQL Execution (see page 44)
SQL Editor (see page 51)

From the CA Database Management Solutions Main Menu

To access ISQL from the CA Database Management Solutions for DB2 for z/OS Main Menu, enter **I** at the command line and press Enter.

The ISQL Online SQL Execution panel appears.

From Any Product Command Line

Enter the jump command ISQL on the command line of any CA Database Management Solutions for DB2 for z/OS product and press Enter.

The SQL Editor panel appears.

From CA RC/Update

To start ISQL from CA RC/Update, enter IS (Interactive SQL) in the Option field of the main header and press Enter. The Control Panel screen appears. Choose a file and the SQL options to be performed from this screen.

From the SQL Editor in Another Product

If you have accessed the SQL Editor from a CA Database Management Solutions for DB2 for z/OS product, enter **ESQL** in the command line and press Enter. The ISQL Online SQL Execution panel appears.

Note: Do not use the ESQL command if the SQL Editor was entered using the ISQL jump command.

From Outside Our Products

From outside the CA Database Management Solutions for DB2 for z/OS, you can start ISQL from a TSO command line or ready prompt, and from an ISPF edit session.

From a TSO Command Line

If you have CA RC/Update, you can start ISQL from any TSO command line outside of the CA Database Management Solutions for DB2 for z/OS, provided that the products are not already running in the TSO session. To do so, specify TSO ISQL in the command line and press Enter.

The ISQL command has optional keyword parameters for specifying the DB2 subsystem ID, the parmlib data set name, and the global parmlib suffix.

More information:

TSO ISQL Command Syntax (see page 38)
Parameter Descriptions (see page 40)

From a TSO Ready Prompt

If a TSO Ready prompt appears, enter ISQL and press Enter. The SQL Editor panel appears.

Note: The ISQL command does not accept any additional parameters when entered from a TSO Ready prompt. By default, it uses the default settings from your last ISQL session, the parmlib data set name from your last session, and the default parmlib suffix.

From an ISPF Edit Session

To access ISQL from an ISPF edit session, you must first identify the block of SQL to test. To mark the first and last lines of the block of SQL, enter EE in the line command area next to the first line and the last line. If you mark statements that are not SQL statements, an SQL error occurs when the statements are submitted for execution.

Note: If you try to mark a block that is larger than one screen, you will receive the error message “Block command not complete” when you try to page down. Type SQL on the command line and press PF8 (down), not Enter. You can scroll through the file and mark the end of the block of SQL to be tested by entering EE.

You can also specify the first line and the total number of lines to be tested by entering En where n is the total number of lines to test.

Note: It is not necessary to be exact with the block or the number of lines as long as the entire SQL statement is present, since ISQL will ignore unnecessary information.

Once the lines to test have been specified, enter SQL on the command line and press Enter to start ISQL. The ISQL Online SQL Execution panel appears. Access the edit screen to make changes to the SQL before execution, press PF3 (or enter the END command) to return to the ISQL Online SQL Execution screen. You are returned to your original position in the ISPF edit session when you enter the SEND command.

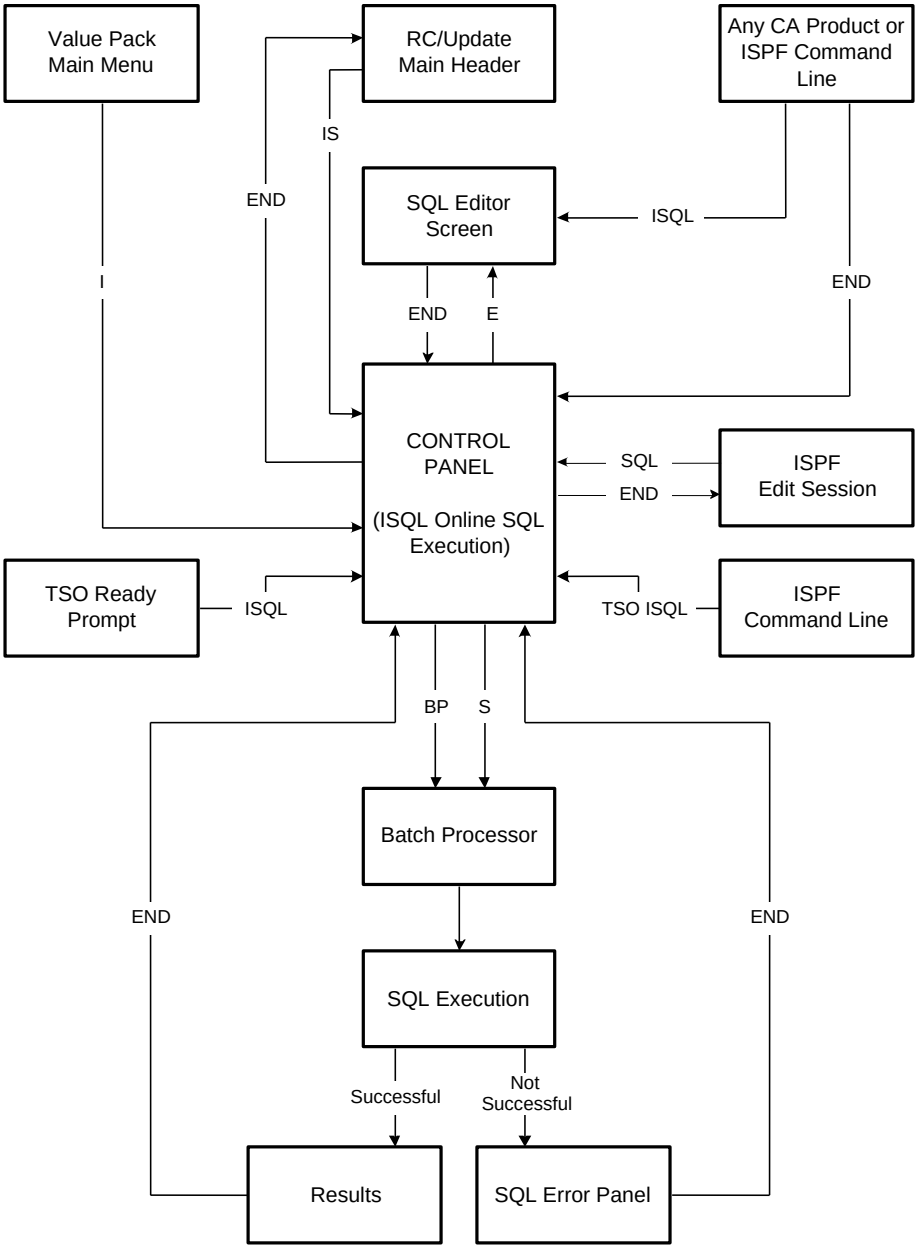
You can also specify optional positional parameters for the DB2 subsystem ID, the parmlib data set name, and the global parmlib suffix with the SQL edit macro.

More information:

Parameter Descriptions (see page 40)

Screen Flow

The following diagram illustrates the screen flow when using ISQL from in an ISPF edit session and from in CA RC/Update.



With the screen flow exceptions noted, ISQL works the same, regardless of whether it was started from an ISPF edit session or from CA RC/Update.

ISQL and SQL Commands

You can use the TSO ISQL and SQL edit macro commands to start ISQL and execute SQL statements contained in a data set.

TSO ISQL Command Syntax

ISQL is a TSO command, and as such, all parameters are entered using keywords. Each of the keywords has a long descriptive name, and a shorter abbreviation as follows:

- SSID or SS
- SUFFIX or SU
- PARMLIB or P

The following syntax shows the long descriptive names:

```
TSO ISQL SSID(ssid) SUFFIX(suffix) PARMLIB(parmlib_data_set_name)
                                   PARMLIB(implicit_parmlib_ddname)
                                   PARMLIB(DD:explicit_parmlib_ddname)
```

The following syntax shows the abbreviated names:

```
TSO ISQL SS(ssid) SU(suffix) P(parmlib_data_set_name)
                             P(implicit_parmlib_ddname)
                             P(DD:explicit_parmlib_ddname)
```

Note the following:

- All parameters are optional.
- The third positional keyword for the parmlib accepts a data set name, an implicit ddname, or an explicit ddname.
- All parameters must be separated by one or more blanks.

More information:

From a TSO Command Line (see page 35)
Parameter Descriptions (see page 40)

Use a Data Set Name

You *must* use the following rules when entering data set names with the SQL or ISQL commands:

- You can only enter a single data set name
- Quotes are ignored

In other words, the data set name is always treated as if it were a fully qualified data set name, regardless of whether it is entered with quotes. This means that '*high-level*.parmlib' and *high-level*.parmlib are both the same; the TSO prefix will not be appended to the latter.

Using the examples in Parmlib Data Set Name, the data set name could have been entered without quotes with the same result:

```
SQL . . high-level.parmlib
TSO ISQL PARMLIB(high-level.parmlib)
TSO ISQL P(high-level.parmlib)
```

Use a DDNAME

Instead of entering a single data set name, you can enter the ddname of a preallocated parmlib data set or multiple data sets. This is the only way to allocate more than one data set.

In the following examples, *high-level*.parmlib1, *high-level*.parmlib2, and *high-level*.parmlib3 are allocated to a ddname called PARMDD.

```
SQL . . PARMDD
TSO ISQL PARMLIB(PARMDD)
TSO ISQL P(PARMDD)
```

Note: The syntax is the same whether you are specifying an implicit ddname or a data set name. This is called an *implicit ddname* specification. If the parmlib parameter is eight characters or less, and contains no decimal points, then it assumed to be a ddname; otherwise, it is assumed to be a data set name, and is treated accordingly.

An *explicit ddname* can be specified using the DD: prefix. Explicit ddnames are not parsed, and are always unconditionally treated as a ddname. See the following examples:

```
SQL . . DD:PARMDD
TSO ISQL PARMLIB(DD:PARMDD)
TSO ISQL P(DD:PARMDD)
```

SQL Edit Macro Syntax

The syntax for the SQL command is identical to the ISQL command; except that the parameters are all positional, no keywords are used. In other words, SSID is the first parameter, SUFFIX is the second parameter, and PARMLIB is the third parameter. When necessary, a period (.) is used as a placeholder to indicate a missing value, see the following examples:

```
SQL subsystem parmlib_suffix parmlib_data_set_name
                             implicit_parmlib_ddname
                             DD:explicit_parmlib_ddname
```

Note the following:

- All parameters are optional.
- The third positional keyword for the parmlib accepts a data set name, an implicit ddname, or an explicit ddname.
- All parameters must be separated by one or more blanks.

More information:

Parameter Descriptions (see page 40)

Parameter Descriptions

You can specify the following optional parameters with the ISQL and SQL commands:

Note: For a complete description of the parmlib data set and the global parmlib suffix, see the CA Database Management Solutions for DB2 for z/OS *Implementation Guide*. Consult your system administrator to determine which parmlib data sets and suffixes are valid at your site.

DB2 Subsystem ID

Specifies the DB2 subsystem to use with the ISQL or SQL command. ISQL automatically connects to the DB2 subsystem used in the CA Database Management Solutions for DB2 for z/OS products session.

Specify the DB2 subsystem ID parameter as the first positional parameter with the SQL command, or as SSID or SS with the ISQL command. For example, to connect to the PT12 subsystem ID using the SQL command, enter the following command:

```
SQL PT12
```

To connect to the PT12 subsystem ID using the ISQL command, enter either of the following commands:

```
TSO ISQL SSID(PT12)
```

```
TSO ISQL SS(PT12)
```

Global Parmlib Suffix

Includes a two-digit number from 00 to 99 at the end of the parmlib data. A default suffix is defined in the parmlib data set itself, and might be different for each installation. Typically, the default suffix is 00. If a suffix is not specified on the ISQL or SQL command, the default is used.

The global parmlib suffix lets you have have multiple versions of the same global parmlib members for different environments. In other words, it lets you create different run-time environments.

Note: Unlike the DB2 subsystem ID and parmlib data set name parameters, the global parmlib suffix is not remembered from one session to the next.

Specify the global parmlib suffix as the second positional parameter with the SQL command, or as SUFFIX or SU with the ISQL command. For example, to specify a suffix of 99 using the SQL command, enter:

```
SQL . 99
```

Note: A period (.) was used in the first position to indicate that the SSID was not entered.

To specify a suffix of 99 using the ISQL command, enter either of the following commands:

```
TSO ISQL SUFFIX(99)
```

```
TSO ISQL SU(99)
```

Parmlib Data Set Name

Specifies the parmlib data set name. By default, ISQL uses the parmlib data set name used in the last session.

Specify the parmlib data set name as the third positional parameter with the SQL command, or as PARMLIB or P with the ISQL command. For example, to specify a parmlib data set name of *high-level*.parmlib with the SQL command, enter:

```
SQL . . 'highlvl.parmlib'
```

Note: A period (.) was used in the first and second positions to indicate that the DB2 subsystem ID and global parmlib suffix were not entered.

To specify a parmlib data set name of *high-level*.parmlib with the ISQL command, enter either of the following commands:

```
TSO ISQL PARMLIB('highlvl.parmlib')
TSO ISQL P('highlvl.parmlib')
```

More information:

From a TSO Command Line (see page 35)
From an ISPF Edit Session (see page 36)

Examples

This section provides examples using the SQL and ISQL commands:

Example 1

Start ISQL, connecting to the DB2 subsystem from your last session, the same parmlib data set, and the default global parmlib suffix:

```
SQL
TSO ISQL
```

Example 2

Start ISQL, connecting to the DB2P subsystem, using the same parmlib data set from your last session, and the default parmlib suffix.

```
SQL db2p
TSO ISQL  SS(db2p)
```

Example 3

Start ISQL, connecting to the DB2P subsystem, using the same parmlib data set from your last session, but this time using the parmlib suffix 03, which is defined by your site.

```
SQL db2p  03
TSO ISQL  SS(db2p)  SU(03)
```

Example 4

Start ISQL, connecting to the DB2P subsystem, specifying a parmlib data set name of *high-level*.parmlib, but using the default parmlib suffix defined in *high-level*.parmlib. Again, note that you can only specify a single data set name.

```
SQL db2p  .  high-level.parmlib
TSO ISQL  SS(db2p)  P(hilevel.parmlib)
```

Example 5

Same as the previous example, except you are preallocating three parmlib data sets to PARMDD.

```
SQL db2p . PARMDD
SQL db2p . DD:PARMDD

TSO ISQL SS(db2p) P(PARMDD)
TSO ISQL SS(db2p) P(DD:PARMDD)
```

Example 6

Start ISQL, connecting to the DB2 subsystem from your last session, a preallocated parmlib data set allocated to PARMDD, and the default parmlib suffix.

```
SQL . . PARMDD
SQL . . DD:PARMDD

TSO ISQL P(PARMDD)
TSO ISQL P(DD:PARMDD)
```

Tips

This section provides some tips for handling some common situations you might encounter when using the ISQL or SQL commands.

ISPF Dialog Errors

If ISQL does not start successfully, an error message appears on the panel where the ISQL or SQL command was issued. If you then press PF1, you might get an ISPF Dialog Error panel with a Panel not found error message.

To determine why the ISPF Dialog error occurred, look at the ISQL and SQL clists. Notice that they have changed dramatically from earlier versions. Specifically, they no longer have any hard-coded allocations. This is because the runtime environment is defined dynamically through a combination of the parmlib data set and the parmlib suffix.

When you enter the ISQL or SQL command, the DSNAMExx parmlib, in which the runtime libraries are provided, gets processed (the runtime libraries are dynamically allocated and ISQL is started).

If an error prevents ISQL from starting during this processing, an error message is generated.

If the message has a help panel associated with it (.HELP=panel) and you press PF1, ISPF is unable to locate the help panel, and displays an ISPF dialog error because all of the runtime libraries (including the HELP library) have been unallocated.

To prevent this from occurring in the future, you should permanently allocate the CA-DB2 help panels to the TSO session's SPFHLIB concatenation.

For example, the most common error message that you might see, is the result of a DB2 connection error, such as: RC=106, RCODE=0000 DB2 SSID XXXX is not defined in the CA-DB2 PARMLIB.

If you press PF1 after this message appears and you have not allocated the CA-DB2 tools help library to your TSO session, you will receive the ISPF dialog error. The solution, in this example, is to enter the correct DB2 subsystem ID with the ISQL or SQL command.

Save the Global Parmlib Suffix from Each Session

The global parmlib suffix is not saved from session-to-session, and must be specified each time.

To save the global parmlib suffix from each session

1. Make a copy of the ISQL or SQL clist, and rename. See Copy the ISQL or SQL Clists for important information.
2. Modify the clist to add a SET statement.

Important! Do not, under any circumstances, change the parameters or default values on the PROC statement.

For example, to avoid always having to type TSO ISQL SUFFIX(37), copy and rename the ISQL clist to ISQL37. Modify it to add a SET statement. After you save your changes, you only have to enter TSO ISQL37.

If you are unfamiliar with clists, or if you are unsure where to make the changes, contact Technical Support for assistance.

Copy the ISQL or SQL Clists

If you copy the ISQL or SQL clists to another library, you must also copy the RSPDEF, RSPINIT, and RSPFREE clists.

ISPF Online SQL Execution

The ISQL Online SQL Execution panel lets you enter control parameters and select SQL options.

From this panel, you can do the following:

- Specify the control parameters and host variables
- Specify an SQL option

- Select the processing mode
- Change the DB2 SSID
- Enter host variable values

The DB2 version appears in the DB2 Version field. The SQL that will be executed appears at the bottom of the panel. If there are many statements, use the scrolling key to view them all.

More information:

Start ISQL (see page 34)

Specify Control Parameters

The Control Parameters on the ISQL Online SQL Execution panel help place controls on data retrieval by SQL SELECT statements (except for COMMIT and ROLLBACK).

To specify control parameters, enter data as applicable in the following fields:

Select Row Limit

Specifies a number to limit the number of rows selected (when an SQL SELECT statement is executed). The default value is 99,999,999.

Max Character Size

Specifies a character number to change the maximum width of columns that appear. The default is 99,999.

The previous fields are ignored for SQL statements except SELECT.

Commit or Rollback

Indicates how the executed SQL is processed. Specify C (commit) or R (roll back). The default is R.

Rollback is especially useful for updates and deletes because it lets users test the SQL, view the results, and then rollback the changes. Tests can be made again and again without changing test data.

Continue if Warnings

Indicates whether execution should continue if non-SQL-level warnings are encountered. If Y, processing continues if non-SQL-level warnings are encountered. This causes the Batch Processor .OPTION ERRORS command to be added to the front of the input data set. The Batch Processor continues executing on non-SQL return codes that are less than or equal to 4. Entering Y will not cause execution to continue on non-zero SQL return codes.

If N, processing does not continue when a non-zero return code is returned. This causes the Batch Processor .OPTION NOERRORS command to be added to the front of the input data set.

Continue if SQL Errors

Indicates whether execution should continue if SQL errors are returned. If Y, processing continues if any SQL errors are encountered. The Batch Processor .OPTIONS SQLERRORS command is added to the front of the input data set. If a large block of code is being marked for execution and a non-SQL code is included in that block, specify Y to continue when errors are returned. This enables testing of the SQL without first deleting non-SQL code.

If N, processing does not continue if a negative SQL code is returned. The Batch Processor .OPTION NOSQLERRORS command is added to the front of the input data set. N is the default.

Output to Dataset

Indicates whether to write the results of an SQL SELECT statement to a sequential data set or a PDS. If Y, when the SQL is executed, the Select Dataset Specification panel appears. If N, the results are not written.

View Audit File

Indicates whether to view the audit file. If Y, the audit file always appears. If N, the audit file only appears if errors occurred.

You can save the control parameter values specified on the ISQL Control Panel in an ISPF Profile by entering the SAVE command from the Control Panel command line. The values are used as defaults for future ISQL sessions.

More information:

Specify a Data Set for Output (see page 50)

Specify Host Variables

Host variables for all selected statements are parsed out and listed in the Host Variable fields on the ISQL Online SQL Execution panel. A value must be entered for each of the host variables for the SQL to execute successfully. If multiple SQL statements were submitted for execution from the ISPF Editor, all host variables will be listed.

- In batch mode, the SQL to be executed will be written to a file with the Host Variable values replaced.
- If there are more than 10 host variables displayed, you can view them using the vertical scrolling keys.
- If there are no host variables, this section will not be displayed.

Use the H (Header) command to toggle off the display of the header, SQL options, and control parameters. Only the host variables and the SQL appear. This is especially helpful if there are many host variables.

Specify SQL Options

Select any of the following SQL options by entering the option letter in the Option field and pressing Enter.

S (SQL Execution)

Executes the SQL.

For SELECT statements, once the execution has completed, data appears for online viewing or a bad return code appears.

For other statements, the resulting return code and message appear.

If non-SQL statements are submitted for execution, an SQL error appears.

D (Data Set I/O)

Imports SQL from a data set or exports SQL to a data set for modification, templating, browsing, or execution.

BP (Batch Processor/Submit)

Executes all DDL saved to a data set.

Note: For more information about the batch processor, see the *CA Database Management Solutions for DB2 for z/OS Batch Processor Reference Guide*.

E (Edit SQL)

Edits the SQL to be executed.

More information:

Import and Export SQL (see page 49)
SQL Editor (see page 51)

Specify the Processing Mode (Online or Batch)

ISQL always starts in online mode. To switch to batch mode, a partitioned data set to which the SQL should be written must be specified. Because ISQL is a separate component, it uses a different data set allocation.

Note: For more information about executing the DDL, see the *CA RC/Update for DB2 for z/OS User Guide*.

To write SQL to a data set

1. Change the Mode parameter to B (batch), and press Enter. The Batch Specification panel appears.
2. Specify a data set and member, and press Enter to return to the ISQL Online SQL Execution panel. If you invoked ISQL using the IS option from CA RC/Update, this data set cannot be the same as the data set specified in CA RC/Update.
3. Enter **S** (SQL execution) for the option to submit the SQL to the data set.
4. Change the mode back to O (online) and close and unallocate the data set.
5. Select the BP option from any CA RC/Update panel to submit the data set to the Batch Processor for execution and press Enter.
6. Enter the name of the ISQL data set to be executed.

Browse Select Results

The rows returned from an SQL SELECT statement executed in Online Mode appear on the Browse Select Results panel.

Notice that the column names are at the top of the screen in uppercase letters. Use the scrolling keys and commands to view all the data. The number of rows returned appears above the column names.

To print the page viewable on the panel, use the PPRINT command. To print the entire report of scrollable data, use the QPRINT command. If the data is wider than the page (print) width, it is truncated.

Note: For more information about scrolling and print commands, see the *CA Database Management Solutions for DB2 for z/OS General Facilities Reference Guide*.

When finished with the Browse Select Results panel, press PF3 (End) to return to SQL Execution. The next statement is executed or processing ends.

Import and Export SQL

SQL can be imported and exported SQL from and to a data set for modification, templating, browsing, or execution. An input data set can be specified from which to retrieve (import) SQL statements and an output data set can be specified in which to save (export) SQL statements. There is also an option to write the results of an SQL SELECT statement to a data set.

To import or export SQL, specify the D (Dataset I/O), in the Option field on the ISQL Online SQL Execution screen. You can also specify the Output to Dataset and View Audit File control parameters.

More information:

Specify SQL Options (see page 47)

Specify a Data Set for I/O

The Data set I/O Specification panel appears when the Data set I/O option is selected from the ISQL Online SQL Execution panel.

To specify a data set for I/O, enter data as needed in the following fields:

Operation

Indicate the operation to perform. Valid values are as follow:

- I-Read contents of input data set into ISQL. If specified, an input data set must be specified in the Input Data Set Name field.
- O-Write contents of ISQL to output data set. If an output data set is not specified in the Output Data Set Name field, the data will be written to the input data set.

Input Data Set Name

Specify an input data set name, including the member name. The member must exist.

To specify a fully qualified name, enter the name in single quotation marks. Otherwise, your TSO ID will be prefixed to the data set name.

Volume Serial

Enter the VOLSER if the data set is not cataloged.

Output Data Set Name

Specify an output data set name, including the member name. To specify a fully qualified name, enter the name in single quotation marks. Otherwise, your TSO ID will be prefixed to the data set name.

If a member is specified that does not exist, ISQL will create it. If an existing member is specified, the current contents are deleted and the new SQL exported.

Volume Serial

Enter the VOLSER if the data set is not cataloged.

When the data sets have been specified, press PF3 (End) and the ISQL Online SQL Execution panel appears. If SQL was imported, the data set contents are displayed in the lower portion of the screen.

Specify a Data Set for Output

The Select Dataset Specification panel appears when the choice is made to execute SQL (if the Output to Dataset option was chosen on the Control Panel). This panel lets you specify a data set to receive the SQL execution output.

Data Set Name

Specifies a flat file data set name. The output cannot be written to a partitioned data set.

Record Length

Specifies the logical record length of the data set.

Block Size

Specifies the maximum length of one block in the data set.

Record Format

Specifies the record format of the data set. Valid values are as follow:

- F-Fixed
- FB-Fixed block
- FBA-Fixed block with ASA character
- V-Variable
- VB-Variable block
- VBA-Variable block with ASA character

Device Type

Specifies the device type on which the data set is located. Use a generic DASD unit name, such as SYSDA, CART or TAPE.

Primary Space

Specifies the primary space quantity, in tracks, for a new data set. This field must be set to a numeric value greater than zero.

Secondary Space

Specifies the secondary space quantity, in tracks, for a new data set. This field must be set to a numeric value of zero or greater. If blanks are entered in this field, it defaults to a value of one.

Note: The Primary Space and Secondary Space fields are used only when a new data set is being created. If the output is to an existing sequential data set, the data set will be recreated using these space attributes. If the output is to an existing partitioned data set, these space attributes are ignored.

If an existing data set is specified, but the user changes the characteristics, ISQL deletes and recreates the data set.

SQL Editor

The SQL Editor panel appears when you do any of the following:

- Enter the E option on a confirmation panel.
- Enter the E option on the ISQL Online SQL Execution panel.
- Enter the jump command ISQL from any TSO command line.
- Create or alter a view using the View option.
- Enter standard ISPF Edit line commands (copy, move, insert, delete, repeat, block commands) at the beginning of any SQL lines to manipulate them.
- Enter the SQL statements or Batch Processor commands to test in the SQL area. Each statement must be separated with a semicolon (;). Use PF7 and PF8 (up and down) to scroll in this area.

When editing is complete, press PF3 (End). Depending on how the editor was accessed, the following results occur:

- ISQL or IS options-The ISQL Online SQL Execution panel appears. Enter the CANCEL command to exit. If ISQL was accessed from in the ISPF editor, any changes made to the SQL using the SQL Editor are not reflected in the original data set.
- RC/Object DDL Confirmation-The Confirmation Screen appears, with changes reflected in the displayed DDL. If returning to the object panel from the Confirmation Screen, the changes are not reflected on the object panel.
- Create, Template, Edit Views-The View Confirmation Panel appears.

Note: For detailed information about the SQL editor commands, see the *CA Database Management Solutions for DB2 for z/OS General Facilities Reference Guide*.

More information:

Start ISQL (see page 34)
Specify SQL Options (see page 47)

ISQL Processing Considerations

When processing ISQL, note the following additional considerations:

- Row limit
- Interrupt processing
- SET CURRENT SQLID command
- Date and time formats
- ISQL threads

Row Limit

There may be a desire to limit the number of rows returned from a SELECT statement to lessen processing time. The default value of the limit of rows selected is 99,999,999. A MAXCHAR size can also be specified to reduce the memory required for SELECT results.

Interrupt Processing

A long running SELECT statement can be interrupted by pressing the ATTN key. This will stop data retrieval and display the Data Retrieval Interrupt screen. This screen shows the number of rows retrieved so far and offers three options:

- Continue retrieval
- Stop data retrieval and browse results
- Stop data retrieval and cancel request

Note: For more information about interrupt processing, see the *CA Database Management Solutions for DB2 for z/OS General Facilities Reference* Guide.

SET CURRENT SQLID Command

When using the SET CURRENT SQLID command, the SET CURRENT SQLID statement is generated and added to the beginning of the selected SQL before it is submitted for testing. This only occurs if the current user ID is different than the primary user ID.

Note: For more information about the SET CURRENT SQLID command, see the *CA Database Management Solutions for DB2 for z/OS General Facilities Reference* Guide.

Date and Time Formats

ISQL and the Batch Processor are precompiled with the DATE(ISO) and TIME(ISO) options. This is required for internal processing. If the SQL created in ISQL (and submit to the Batch Processor) contains references to DATE/TIME values or functions, and the DATE/TIME format specified in DSNHDECP is not ISO, unexpected results can occur. In this case, since the ISO formats are built in to the CA products, the DATE/TIME format may have to be explicitly included in the SQL, to ensure the results are correct.

ISQL Threads

When ISQL executes multiple SQL statements on the same table, a -911 situation can occur if one of the statements is an update. This is because ISQL takes a SHARE LOCK on the object until you enter PF3. To prevent this, specify THREADS(2) in the PARMLIB(BATPROC) or insert a COMMIT statement between the SQL statements.

Chapter 5: Using Alternate Catalog Mapping (ACM)

This chapter describes how to create and maintain alternate catalog maps, assign alternate names, and create alternate catalog map shadow tables that define the options for loading and storing the shadow tables. It also describes how to upgrade ACM IDs, create views, and activate ACM.

This section contains the following topics:

Alternate Catalog Mapping

The CA Database Management Solutions for DB2 for z/OS let you view DB2 catalog information without SYSADM authority. The Alternate Catalog Mapping (ACM) facility reduces catalog contention and eases security concerns by redirecting catalog queries to views defined over the system catalog tables or shadow tables. This lets you view DB2 catalog information without having SELECT privilege on the actual catalog tables and improves performance.

You can create the alternate catalog maps for individual users or groups of users and then those users can use the shadow tables and views to generate reports.

After you define your alternate catalog mappings and copy data into the shadow tables and views, the ACM facility lets you do the following:

- View DB2 catalog information without having SELECT privilege on the actual catalog tables.

  Without ACM, you must grant SELECT authority on all catalog tables to PUBLIC or to each user that needs catalog information. This is a tedious and risky task from a security administration perspective. With ACM, users without SELECT authority on the catalog tables can generate reports using the information you provide through shadow tables and views.

- Improve performance by reducing contention on the catalog tablespaces.

  When shadow tables are used to provide results from catalog queries, only the tablespace containing the shadow tables is locked. Furthermore, the user requesting catalog information from the shadow tables does not need to wait for other system catalog access to complete.

  You can further improve performance of catalog queries by defining additional indexes on the shadow tables. Any combination of shadow tables and views can be used in an ACM definition so that you can adjust the performance, resource consumption, and security management improvements based upon your needs.

ACM Creation and Maintenance

You can create an alternate catalog mapping (ACM) for a single user or for multiple users. You can also update, copy, and delete ACMs as needed.

Create an ACM for a Single User

To create an ACM for a single user

1. Type **M** in the Option line or select Alt Catalog Mapping from the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The Alternate Catalog Mapping Services panel appears. Each row represents one alternate catalog map to the shadow tables created for use with ACM.

   Note: To view a specific list of ACM IDs, enter search criteria in the List for ACMID header field.

2. Complete the following fields:

   ACMID

   Specifies the name you want to call the ACM.

   Note: To create a default ACM for all users, specify DEFAULT as the name of the ACM. The default is used instead of the actual system catalog tables when ACM is set to ON and an ACM ID has not been defined.

   Description

   (Optional) Specifies a description of the new ACM.

   Press Enter.

   The Alternate Name Assignment panel appears.

3. Specify a creator ID in the To Creator field and an equal sign (=) next to the table name in the To column and press PF3.

   The catalog mappings are assigned and the Alternate Catalog Mapping Services panel reappears.

   Note: You can also copy from one ACM ID to another or from one group to another.

Create an ACM for Multiple Users

You can assign multiple user IDs to the same set of shadow tables by entering wildcard characters (% or _) for the ACMID and then selecting user IDs from the selection list.

To create an ACM for multiple users

1. Type **M** in the Option line or select Alt Catalog Mapping from the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The Alternate Catalog Mapping Services panel appears.

2. Complete the following fields:

   ACMID

   Specifies the name you want to call the ACM with a wildcard character. You can use the percent sign (%) or underscore (_) wildcard characters in the ACM name as follows:

   Percent sign (%)

   Indicates that zero or more characters can occupy that position and all remaining positions to the end of the name or to the next character. For example, USER% will retrieve all names that start with USER. USE%06 will retrieve all names that start with USE and end with 06.

   Underscore (_)

   Indicates that any character can occupy that position. You can repeat this character at multiple locations in the name of the ACM.

   Description

   (Optional) Specifies a description of the new ACM.

   Press Enter.

   The USERID Selection List panel appears. A list of user IDs that match the selection criteria appears. You can change the criteria by entering new selection criteria in the User ID field.

3. Type **S** in the SEL column to select a user or use the SS block command to select a group of users and press Enter.

   The USERID Selection List panel reappears showing only the selected users. Repeat as needed until you have added all users you want to the ACM.

   Note: To remove a user ID from the list, type U next to the user ID in the SEL field and press Enter. To review the list of selected users, type S in the Show field.

4. Review the selection list and press F3 (End) to process.

   The Alternate Name Assignment panel appears.

   Note: The Define ID display field contains MULTIPL if you selected more than one user ID.

5. Type **&ACMID** in the To Creator field and an equal sign (=) in the To table name field to assign alternate catalog mappings for multiple users and press F3 (End).

   The Alternate Catalog Mapping Services panel reappears. New or updated catalog mappings are assigned to all user IDs you specified. The ACMID for each user is the same as their user ID.

Add an ACM Using an Existing Map

You can add an ACM using an existing map.

To add an ACM using an existing map

1. Type **M** in the Option line or select Alt Catalog Mapping from the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The Alternate Catalog Mapping Services panel appears.

2. Type **S** (select) in the Cmd field next to the ACM ID you want the ACMs copied to and press Enter.

   The Alternate Name Assignment panel appears.

3. Type **C** (copy) in the Cmd field next to the ACM ID you want the ACM's copied from and press Enter.

   The Alternate Name Assignment panel reappears.

4. Specify an ACM ID in the Copy from ID field to specify the source of mappings and press Enter.

   Note: The alternate catalog mappings for that ACM ID will be automatically entered in the Creator and Table Name fields of the TO scrollable area.

The following message appears:

`VALUES USED FROM COPYID` *name*

name

Identifies the ACM ID whose mappings you have copied.

5. Press F3 (End).

   The catalog mappings are assigned and the Alternate Catalog Mapping Services panel reappears.

Update an ACM

You can update an alternate catalog map.

To update an ACM

1. Type **M** in the Option line or select Alt Catalog Mapping from the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The Alternate Catalog Mapping Services panel appears.

2. Type **S** (select) in the Cmd field next to the alternate catalog mapping assignment you want to change and press Enter.

   Note: You can also optionally change the Description field.

   The Alternate Name Assignment panel appears.

3. Update the catalog mappings and press F3 (End).

   The catalog mappings are updated and the Alternate Catalog Mapping Services panel reappears. If you are only updating the ACM description, no changes appear on this panel.

Use an Existing ACM as a Template for Another ACM

You can update an alternate catalog map using an exisiting map.

To use an existing ACM as a template for another ACM

1. Type **S** (select) on the command line area for the ACM ID you want the alternate catalog maps copied to, and **C** (copy) on the command line area for the ACM ID you want the alternate catalog maps copied from, and press Enter.

The Alternate Name Assignment panel appears. The copied alternate catalog map's values are entered for the selected ID and the following message appears:

`VALUES USED FROM COPYID` *name*

name

Identifies the maps copied from the ACM ID you selected.

2. Press F3 (End).

The catalog mappings are assigned and the Alternate Catalog Mapping Services panel reappears.

Delete an ACM

To delete an alternate catalog map

1. Type **M** in the Option line or select Alt Catalog Mapping from the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The Alternate Catalog Mapping Services panel appears.

2. Type **D** (delete) in the Cmd field next to the alternate catalog map you want to delete and press Enter.

   The map is deleted and the status changes to DEL.

Assign Alternate Catalog Mappings for a Single User

Alternate catalog mappings for a single user can be assigned.

To assign alternate catalog mappings for a single user

1. Type **M** in the Option line or select Alt Catalog Mapping from the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The Alternate Catalog Mapping Services panel appears.

2. Select an ACMID and press Enter.

   The Alternate Name Assignment panel appears.

3. Specify the ACM ID in the Copy from ID field and press Enter to assign alternate catalog mappings for a user (from one ACM ID to another).

   The following fields are updated on the panel:

   - The alternate catalog mappings for that ACM ID are automatically entered in the table name and creator fields of the TO scrollable area. Once you press Enter, this field is cleared.
   - The From field shows the group of DB2 catalog tables for which you are defining alternate names. The first part of the name, SYSIBM, is the creator ID; the second part represents the table name. (The list of catalog names appears in the scrollable area directly under this field).

4. Specify the creator of the alternate catalog table or view in the To Creator field and press Enter.

   The specified value is carried down to the individual table creator fields in the TO scrollable area. This field often represents a department or other group of users. You can enter an equal sign (=) to copy the ACMID to the creator fields in the TO scrollable area.

5. Specify the table name in the To Table Name field and press Enter. This part of the name is most often the same as the actual catalog table. You can also enter an equal sign (=). The names of the IBM catalog tables are listed in the FROM scrollable area. SYSIBM is the creator ID (the first part of the table name). The second part of the name is the actual catalog table name.

   The specified value is carried down to the individual table name fields in the TO scrollable area.

6. Specify the alternate tables or views that you want to use in place of the catalog tables in the TO scrollable area. The first part of the name represents the creator ID. The second part of the name is the alternate catalog table name. You can enter creator IDs and table names in this area.

   The values shown in the Creator ID and Table Name fields in the TO Scrollable Area are those used for the alternate catalog mapping.

   Note: Remember that shadow tables or views are created outside of the Alternate Catalog Mapping facility. For instructions on ways to create views through CA RC/Update or CA RC/Migrator, see the user guides for these products.

Assign Alternate Names That Match the DB2 System Catalog Names

You can assign alternate names where the alternate table/view names are the same as the DB2 system catalog table names.

To assign alternate names that match the DB2 system catalog names, complete the following fields:

- Specify a creator ID in the To Creator field. This creator ID is automatically entered for all creator IDs in the To scrollable area.
- Enter an equal sign (**=**) in the To table name field to keep the same table names as the SYSIBM catalog table names.
- Optionally, change individual creator or table names by entering the new information in the Creator and Table fields of the To scrollable area.

Press F3 (End).

The catalog mappings are assigned and you are returned to the Alternate Catalog Selection panel.

Create ACM Shadow Tables

You can create Alternate Catalog Mapping (ACM) shadow tables and indexes, and you can define options for loading and storing the shadow tables.

Note: If you use the AUTOSIZE command to calculate the tablespace sizes, we recommend that you run RUNSTATS on the tablespaces. When executed, RUNSTATS updates the SYSIBM tables with the current object sizes and prevents possible space errors when loading the data from the DB2 catalog tables into shadow tables. AUTOSIZE is only available when creating shadow tables on DB2 9. For DB2 10 and above, the AUTOSIZE command is replaced with the DB2 managed data set approach. DB2 handles data set extend sizing more efficiently, which should result in better space management.

Follow these steps:

1. Type **M** in the Option line or select Alt Catalog Mapping from the CA Database Management Solutions for DB2 for z/OS Main Menu and press Enter.

   The Alternate Catalog Mapping Services panel appears.

2. Type **MAKETAB** on the command line (or M next to the ACM definitions you want to make) and press Enter.

   The ACM Shadow Table Creation panel appears.

   Note: This command can also be entered from the ACM Update panel.

3. Specify the shadow table attributes, and press Enter.

   The shadow table attributes are defined and the following processing occurs based on the DB2 version you are running.

   - For DB2 9 and below, the ACM Tablespace Sizes panel appears. Go to Step 4.
   - For DB2 10 and above, the Batch Processor Interface panel appears. Go to Step 5.

4. Specify the primary and secondary space allocations for the shadow tables that hold all data that currently resides in your catalog, and press F3 (End).

   Note: Specify AUTOSIZE in the Option field and press Enter to calculate these values automatically. Complete all panels that appear.

   Important! We recommend that you execute PDASTATS or RUNSTATS on all tablespaces before using this command.

   The space allocations are defined.

5. Change the batch processing options as needed and submit for execution.

   Note: For more information about the Batch Processor options, see the *CA Database Management Solutions for DB2 for z/OS Batch Processor Reference Guide*.

   The shadow table is created.

Upgrade Considerations

When converting from one version or mode of DB2 to an upgraded version or mode, it might be necessary to use the MAKETAB command to create ACM tables that include all of the new tables and columns in the catalog.

This processing also applies to ACM views. If the DB2 version or mode has changed and the catalog tables have been updated with new columns, we recommend that you drop and recreate the dependent views.

Create Views Phrase

You might see the phrase “create views” on the DB2 system catalog tables instead of shadow tables. Views access the same table data, so you always access the most current information. In addition, views provide the option of controlling the information that a user can view and can be used with ACM to reduce the amount of resources that are used to perform certain functions from in the products.

CA RC/Update and CA RC/Migrator provide an easy means of creating these views. Use the View Create feature of these products to create views that include all columns for all tables. Keep the same table name for the view, changing only the creator ID. (This changes the fully qualified names, while keeping the views easy to identify.) For example, change the creator from SYSIBM to VIEW. This would change the DB2 catalog table name SYSIBM.SYSTABLES to VIEW.SYSTABLES.

Note: If you use views with WHERE clauses with ACM, note that SQL errors can occur if dependent objects specified by the WHERE clauses are *not* included. For objects with RI relationships, this can cause problems with referential data integrity.

CA RC/Query provides an easy way to restrict users to viewing only objects that they create.

Activate ACM

Once you have created alternate catalog mappings through the ACM facility, you must tell your product to use these mappings.

To activate ACM, set the ACM option on the CA Database Management Solutions for DB2 for z/OS Main Menu to ON and press Enter. ON is the default value for this option.

ACM is activated. Once ACM is activated, the process attempts to use the alternate system table. However, if this table is not found, the IBM system table is used instead. This means that you could have a mix of ACM and system tables in the process.

Important! If you have created alternate catalog mappings, but do not activate ACM, the product still accesses the DB2 system catalog tables.

If the ACM option is set to OFF, the IBM catalog tables are used.

CM is activated. If you have activated ACM, but have not assigned alternate catalog mappings, the product will use the actual system catalog tables. If the ACM default is set to OFF, the actual catalog tables are used. If you have created alternate catalog mappings, but do not activate ACM, the product will still try to access the DB2 system catalog.

If you have not assigned alternate catalog mappings, specify OFF in the ACM field because ACM enabled products will use static SQL, which is more efficient than dynamic SQL.

Note: The ACM field on the CA Database Management Solutions for DB2 for z/OS Main Menu is the default ACM setting for CA RC/Query and CA Plan Analyzer. To enable or disable ACM through CA RC/Query or CA Plan Analyzer for the current session, use the ACMON and ACMOFF primary commands.

You can create a default Alternate Catalog Mapping for users who do not have Alternate Catalog Mapping. This is an easy way to assign many users to a single default.

Chapter 6: Using the Utility Manager

This section contains the following topics:

Utility Extracts

Utility Manager provides a convenient and efficient facility for submitting on demand utility requests. You can execute any combination of DB2 utilities, third-party programs, and user applications for any selected set of DB2 objects. You can also use line commands to access DB2 utilities directly (IBM and CA).

The Utility Manager lets you create and submit on demand, one-time utility runs using utility extracts. A utility extract is created by the Utility Manager and defines which DB2 objects will have utilities and/or Action Procedures executed against them on a random basis. This enables you to execute utilities such as REORG and STOP DATABASE quickly.

The utility extract contains all the selected DB2 objects and utilities and cannot be saved. Its purpose is to temporarily save your current utility definition. You can only SUBMIT or CANCEL a utility extract; if you decide to SUBMIT, you can request that the resulting job be written to a data set.

To create a utility extract, use the Build Utility Extract screen. From this screen, you can do the following:

- Specify object selection criteria and request utilities.
- View currently selected objects or utilities using the explode option.

Utility extracts are different from Extract Procedures because they let you perform ad hoc, one-time executions. Extract Procedures enable you to rerun executions against previously-defined objects and can be reused until deleted or dropped from CA Database Analyzer.

How To Execute Utilities Using the Utility Manager

The Utility Manager lets you submit on demand utility requests. You can execute any combination of DB2 utilities, third-party programs, and user applications for any selected set of DB2 objects. You can also use line commands to access IBM and CA Database Management Solutions for DB2 for z/OS utilities directly.

Note: References to Action Procedures refer to the CA Database Analyzer Action Procedures facility. You must have a licensed copy of CA Database Analyzer to use Action Procedures. See Action Procedures for more information.

You can choose between two selection paths to indicate the best method for your utility request: Quick and Standard.

- The Quick Selection Path lets you quickly create utility extracts that require a one-time selection of DB2 objects, or does not require a mix of DB2 objects from different databases.
- The Standard Selection Path lets you select DB2 objects based on different sets of selection criteria that you define. You can use this ability to create a "mix" of DB2 objects for your utility request that you would not be able to through only one set of selection criteria.

Each selection path provides you with the functionality you need to best create your requests.

To execute utilities using the Utility Manager, follow these steps:

1. Select DB2 objects. You can select objects at any level and repeat as many times as needed.

   The Autobuild feature is active in the Utility Manager. The Utility Manager automatically creates all necessary utility control statements for each selected object.

2. Select utilities (mix any combination utilities). You can repeat as many times as needed.

   An executable job is generated (all necessary JCL) and placed into a temporary memory data set for your review or submission.

3. Submit the job for execution or write it to a data set for future review and submission.

How To Access the Utility Manager

You can access the Utility Manager using the following methods:

- Type **Y** (Utility Manager) in the Option field on the CA Database Management Solutions for DB2 for z/OS for DB2 for z/OS Main Menu.
- Type **UTIL** on the command line on any CA Database Management Solutions for DB2 for z/OS product panel.
- Type **UTIL** next to an object in a select list on a product panel.

  Note: The object names are retained from the line where the UTIL command was entered. For example, if you type UTIL next to a tablespace name on a Tablespace Table report and press Enter, the Build Utility Extract panel appears with the tablespace and database names filled in the object selection list criteria area.

- Use cursor select from CA RC/Query.

  Note: A valid license for CA RC/Query is required.

- Type **5** (Utility Manager) in the command line on the CA Database Analyzer Main Menu.

  Note: A valid license for CA Database Analyzer is required.

Press Enter.

The Build Utility Extract panel appears.

Cursor Select

Use the cursor select feature of CA RC/Query to access the Utility Manager from an object selection list or report. Cursor select automatically enters the name of the selected object in the appropriate object field and asterisks in the other object fields of the Build Utility Extract panel. The Extract Data Selection report is automatically executed.

Cursor select is valid only when a database, tablespace, or indexspace scrollable report appears.

Note: A valid utility manager object type must be entered (DB, TS, I, or IX). For example, while viewing a Tablespace Table report, you can use the cursor select feature to enter the name of a tablespace in the Tablespace field of the Build Utility Extract panel, enter asterisks in the other object fields, and automatically execute and display the Extract Data Selection report.

Use Cursor Select

You can use cursor select to access the Utility Manager and an Object Selection List.

To use cursor select, complete the following fields on the Build Utility Extract panel:

1. Specify the new value in the DB2 Object field to change the DB2 object type. Valid objects are database, tablespace, and index only.
2. Type **UTIL DB** (for database), **UTIL TS** (for tablespace), **UTIL I** (for indexspace), or **UTIL IX** (for index) in the command line, but do not press Enter.
3. Place the cursor over the name you want entered in the Item Name field.

Press Enter.

The name is inserted in the appropriate object field of the Build Utility Extract panel and the Extract Data Selection report is executed and displayed. When you have completed your work with the Utility Manager, press PF3. You will be prompted back to the report.

Access Utility Manager with Cursor Select From a Report

You can access the Utility Manager with cursor select from a report.

To access the Utility Manager with cursor select from a report

1. Type **UTIL TS** in the command line.
2. Place the cursor on the tablespace name **DSN8S71D.**

Press Enter

The Extract Data Selection panel for tablespace DSN8S71D appears.

Screen Types

The following sections summarize the screens that display when you define a Utility Extract. See the Utility Extract Flow Chart when reviewing this section.

Build Utility Extract

The Build Utility Extract panel is the control center for creating the Utility Extract. It is used to specify the object selection criteria and for requesting utilities. From this screen, you can view currently selected objects or utilities by using the Explode option.

If you enter the UTIL command to access the Utility Manager while viewing an CA RC/Query database, tablespace, table, or index report, you see the following:

- The selection criterion entered in the Item Name field is automatically entered in the appropriate object field.
- Asterisks are entered in the other object fields of the Build Utility Extract panel.

Make any necessary changes to the object selection information on the Build Utility Extract panel and press Enter to display the Extract Data Selection panel.

The Build Utility Extract panel is bypassed when you access the Utility Manager using the cursor select feature.

Exit the Build Utility Extract panel by submitting the defined Utility Extract or by canceling the process.

Extract Data Selection

After you enter object selection criteria on the Build Utility Extract screen, you are presented with the Extract Data Selection screen, which contains a list of DB2 objects that satisfied your search criteria.

The breakdown of the selection list is based on your initial selection criteria. Databases, tablespaces, and tables are *always* shown (regardless of your selection).Because index retrieval typically consumes significant amounts of processing time, indexes do not display unless you specifically request them. To select indexes, you must enter selection criteria in the Index field on the Build Utility Extract screen.

If you select an object at the database level, all related objects are automatically included for the specified utilities. If you also specify selection criteria for indexes, all the database's indexes are also included.

For example, if you enter selection criteria for indexes, select database PTDEMO1, and select the RUNSTATS utility, RUNSTATS is executed for every tablespace and indexspace that resides in database PTDEMO1. Nothing more is required to submit one or more utilities for all objects related to a database.

If you use the cursor select feature to access the Utility Manager from a CA RC/Query report, the Extract Data Selection screen is automatically displayed for the selected object.

Extract Data List

The Extract Data List screen displays the current DB2 objects selected for the Utility Extract. It appears by entering the EXPLODE option on the Extract Data Selection or Build Utility Extract screens.

You can remove selected objects from the Extract Data List screen.

Build Utility Procedure (Y Utility Option)

The Build Utility Procedure screen lets you select utilities for the Utility Extract. It appears when you enter a Utility Option of C (Create) on the Build Utility Extract screen.

A temporary utility procedure name is created for every invocation of this screen. The current list of utility procedures can be viewed using the E (Explode) option, detailed in the following section.

Extract/Action List (E Utility Option)

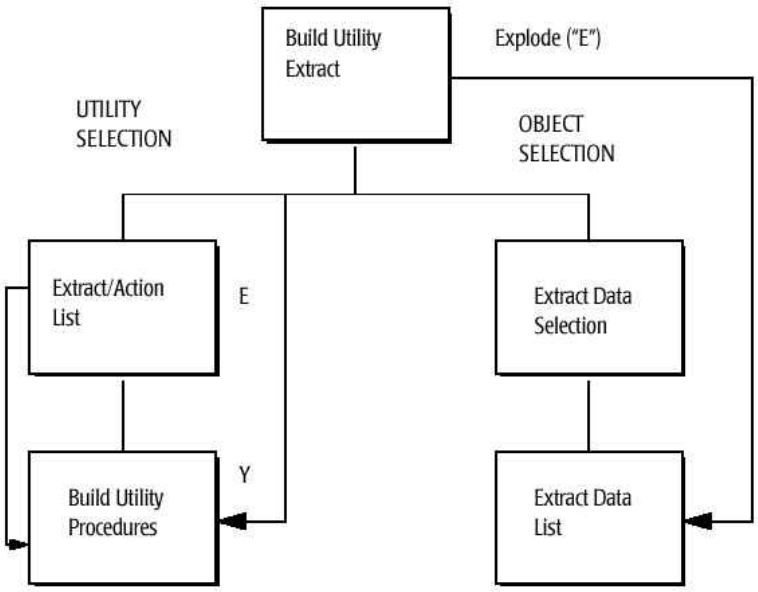
The Extract/Action List appears when you enter a Utility Option of E (Explode) on the Build Utility Extract screen. It is the same screen used to display the connections between Extract/Action Procedures.

The Extract/Action List displays the current list of selected utilities and actions. It lets you delete selected utilities and actions, update the utility and action procedure, or explode the utility and action to see its current definition.

Note: You must own CA Database Analyzer to use Action Procedures. See Action Procedures for more information.

Flow Chart

A possible Utility Manager processing flow is depicted in the following illustration:



If you own CA Database Analyzer, the processing flow might be slightly different. See Action Procedures for more information.

Building Utility Extracts

After you access the Utility Manager, the Build Utility Extract screen appears (unless it is skipped, as explained in the previous sections).

The Build Utility Extract screen is used to create, view, update, and submit Utility Extracts. It is the *control center* of the Utility Manager.

Build Utility Extract

Once the Build Utility Extract screen appears, the steps for Standard Selection Mode are as follows:

1. Enter Object Selection List Criteria as many times as needed.
2. Select desired Generate Utility Option as many times as needed.
3. Enter Submit in the command line to generate and/or submit JCL.

Extract Selection Information

The Database, Tablespace, Table, and Index fields control the objects that will be displayed on the Extract Data Selection screen. The object fields accept standard SQL selection criteria. For more on extract selection, see Extract Selection.

Subsystem and Location Information

Enter information about the subsystem and location that contain the objects that you want to select.

Subsystem Connection ID

Enter the ID of the DB2 subsystem that contains the objects you want to select.

This is the subsystem of the initial DB2 connection if you will be using remote access.

Location

(Default is LOCAL) Enter the name of the location that contains the objects you want to select. The location ID will be displayed on all other Utility Manager screens on the same line as the current user ID.

Generate Utility Option

This prompt controls the selection of utilities for the Utility Extract. Valid options are:

- C-Create. Create a list of utilities for this utility procedure. The Build Utility Procedures screen appears.
- E-Explode. View the current utilities that are presently tied to this Utility Manager session. The Extract/Action Data List screen appears.
- S-Select. Tie an existing Action Procedure to this extract. The Extract/Action Data List screen appears.

Extract Selection

The following list explains the basic rules about using Extract Selection:

- You must select objects before you can select utilities.
- After you select objects, a message is shown requesting that you press Enter to select utilities.
- You can repeat the object selection process with different selection criteria until you get the desired mix of DB2 objects.

- If you enter object selection criteria and a Utility Option the object selection request has priority and the Utility Option is ignored.
- The utilities selected are applied to each object selected (where appropriate).
- If a selected utility does not apply to a selected object type, it is ignored for that object type.
- You can review the current list of selected objects by entering the EXPLODE command in the command line.
- After you have entered a DB2 subsystem ID, the field is protected.
- You cannot create a Utility Extract that references multiple DB2 subsystems.

Selecting Indexes

All object types, except indexes, are always displayed on the Extract Data Selection screen regardless of your object selection criteria. For object prompts without selection criteria, the Utility Manager automatically assumes Asterisk * (all). For example, if you enter PTDB*%* for the database name and leave all other object prompts blank, the Utility Manager automatically displays all tablespaces and tables for any database that begins with PTDB%.

If you want to select indexes, you *must* enter selection criteria for the Index prompt. Indexes are not displayed unless they are requested, because index retrieval consumes a lot of processing time. Enter an *asterisk* (*****) to select all indexes.

Processing

This section describes the processing options available:

- Enter-Invokes (in order of priority):
  - The Extract Data Selection screen if you have entered object selection criteria in the Extract Selection section of the screen.
  - The appropriate utility screen if you have entered a Utility Option.
  - No processing (remain on the screen).
- End-If you have entered information, PF3 (End) has no effect. You must SUBMIT the Utility Extract or CANCEL the process. If you have entered no information, PF3 (End) returns you to the previous screen.
- CANCEL-Enter the CANCEL command to abort the Utility Manager and return to the screen from which you invoked the Utility Manager.

- SUBMIT-When you are ready to submit the Utility Extract, enter the SUBMIT command. The SUBMIT screen for Utility Extracts will be presented.
- EXPLODE-Enter the EXPLODE command to view the selected objects for the Utility Extract.

Note: If you want to view the selected utilities, enter the **E** option for the Utility Option prompt.

Selecting Objects

The Extract Data Selection screen is used to select objects for the Utility Extract. Enter object selection criteria on the Build Utility Extract screen to invoke the Extract Data Selection screen. It also displays when the cursor select feature is used to access the Utility Manager.

Autobuild

The Autobuild option automatically includes all objects dependent on the selected objects. If indexes are displayed, they are also included.

Indexes are displayed on the Extract Data Selection screen and are included with the Autobuild option only if you entered selection criteria in the Index field of the Build Utility Extract screen.

When selecting databases, you must use the A (autobuild) option. All related objects (except indexes) under the database are automatically selected. The A option is very helpful when running utilities against all objects related to a databases.

When selecting tablespaces, tables, and indexes, you can use the A (autobuild) or S (select) option.

Extract Data Selection Screen Layout

The Extract Data Selection screen is designed to provide a hierarchical view of object dependencies based on your initial selection criteria.

Following is a description of the fields on the Extract Data Selection screen:

Scroll

Determines the scrolling amount. CA scrolling is fully supported in the Utility Manager. You can use the FIND command to locate text strings and the QPRINT command to print selection lists. All the scrolling commands except SORT are valid.

PROCEDURE, DESCRIPTION, DATABASE, TABLESPACE, APPLICATION PLAN, TABLE, INDEX

Identifies the header fields. These values were specified on the Build Utility Extract screen. These fields are protected.

The procedure name and description at the top of the screen will always be Utilext-Utility Extraction for a Utility Extract.

O (Option)

Specifies a value to select an object. Specify A (autobuild) to automatically include all objects under the selected object in the utility extract. Specify S to select dependent objects from a selection list.

Database

Specifies the database names.

Tablespace

Specifies the tablespace names. These are indented under the name of the database that contains them. For example, tablespace DSN8S71D is contained in database DSN8D71A. If a tablespace has partitions, the total number of partitions is shown after the tablespace name. If you entered table selection criteria, the individual tablespace partitions are also displayed under the parent tablespace name.

You can select individual tablespace partitions for the Utility Extract by selecting the partition with the S option.

Table

Specifies the table names. These are indented under the name of the tablespace that contains them. For example, tablespace DSN8S71R contains the tables EACT, EDEPT, and EEMP.

Creator

Specifies the creator of the table or tablespace. If you selected only databases, this field contains the creator of the database.

Index

Specifies the index names. These are indented under the name of the table on which they are defined.

Selection Information

This section describes how to select objects for a Utility Extract.

Extract Data Selection: Selecting Databases

When selecting databases, you must use the A (autobuild) option. All related objects under the database will be selected. Indexes will not be included if you did not enter a selection value for the Index prompt on the Build Utility Extract screen. The A option is very convenient when running utilities against all objects related to a databases.

Extract Data Selection: Selecting Tablespaces

When selecting tablespaces, you can use the A (autobuild) or S (select) option. If you enter **A**, all objects related to the tablespace are included in the Utility Extract.

Extract Data Selection: Selecting Tables

When selecting tables, the related tablespace is automatically included.

If you enter **S**, only that table is selected. If you enter A for a table and indexes are displayed, all indexes for that table is included in the Utility Extract. This is an easy way to run utilities on all indexes for a table.

Processing

This section describes the various processing options available:

- Enter-Press Enter to add the selected objects to the Utility Extract and return to the Build Utility Extract screen.
- End-Press End to return to the Build Utility Extract screen *without* adding the selected objects to the Utility Extract.
- CANCEL-Use the CANCEL command to ignore object selections and return to the Build Utility Extract screen.
- EXPLODE-The EXPLODE option displays the current list of selected objects (not including objects currently selected on the Extract Data Selection screen).

View Selected Objects

The Extract Data List screen appears when the Explode option is entered on the Build Utility Procedure or Extract Data Selection screen. The Extract Data List screen shows currently selected objects. You can also delete objects currently contained in the Utility Extract. (The DB2 object is not deleted, only its reference to the Utility Extract.)

Extract Data List Screen Layout

This section discusses the layout of the Extract Data List screen. An Extract Data List screen is shown in the following illustration:

Scroll

Determines the scrolling amount. Scrolling is fully supported in the Utility Manager. You can use the FIND command to locate text strings and the QPRINT command to print selection lists. All the scrolling commands except SORT are valid.

PROCEDURE, DESCRIPTION, CREATOR, SHARE OPTION, DB2 SYSTEM ID

Lists the procedure information is shown for your reference. For Utility Extracts, the procedure name, and description at the top of the screen is always UTILEXT - UTILITY EXTRACTION.

O (Option)

Specifies the action to be performed. To delete an object from the extract list, enter D next to the object names.

Processing

The following processing options are available:

- Enter or End-Press Enter or End to process all delete requests and return to the calling screen.
- CANCEL-The CANCEL command removes all options.

Select Utilities

After you have selected objects for the Utility Extract, you are ready to select utilities. The only requirement for selecting utilities is that you must have already selected at least one object.

You can change your object selection at any time by entering new object selection criteria (on the Build Utility Extract screen) or by deleting previously selected objects using the Extract Data List screen.

This section discusses the creation of utility procedures. A utility procedure is created when you enter the **C** (create) Utility Option on the Build Utility Extract screen.

Build Utility Procedure

You can use the Build Utility Procedures to create temporary utility procedures. Creating a utility procedure is similar to creating an Action Procedure, except that a utility procedure cannot be saved, and therefore has no name or other identifying attributes. A temporary name is automatically assigned to the utility procedure by CA Database Analyzer. The name begins with UTIL and ends with a four-digit sequence number. A new name is created for each “saved” invocation of the Build Utility Procedure screen.

Utility names are displayed on the Extract/Action List screen. The Extract/Action List screen is shown by entering E (Explode) in the Utility Option prompt of the Build Utility Extract screen.

Notice the following terms on the previous screen:

OPT

Views or updates the existing parameters for a specified utility. Possible values are:

- U-Update the parameters for a specified utility.
- E-Explode (view) the current definition of the utility.
- I-Insert utility code.
- M-Move utility code.
- C-Copy utility code and its defined symbolic variables.
- R-Repeat utility code and its defined symbolic variables.

CODE

Specifies utility codes in this field. You can specify up to ten utilities per procedure. Some utilities may access intermediate screens while other utilities access the Model JCL Substitution screen.

SYM (Symbolics)

Specifies the status of symbolic variables. The Utility Manager automatically displays a value in the Symbolics field. This value reflects whether the utility references model JCL that contains symbolic parameters. Possible values are:

- Y-The specified utility references model JCL that contains user symbolic variables.
- D-The specified utility references model JCL that contains default (automatic) symbolic parameters.
- I-The specified utility references invalid model JCL. The referenced member was probably changed after the Action Procedure was saved. You need to update the symbolic variables for the utility.
- N-The specified utility references model JCL that contains no symbolic variables.

UTILITY CODES

Specifies the utilities you want to include in the procedure. See Utility Codes for a list of valid utility codes.

Member Name

Specifies the general DB2 utility model. This model specifies the step JCL statements that are used with every IBM utility code selected for the Action Procedure.

Process Option

Specifies processing options for the model member. Possible values are:

- Y-Indicates that the member is specified.
- U-Enter U to request an update. Using this option displays the PDA Model JCL Substitution screen where you can change any of the user-defined symbolic variables in the specified model JCL member.
- E-Enter E to explode (view) the model JCL member's symbolic variable values.

Library Name

Specifies the PDS library that contains the model JCL member for this procedure.

Note: If you do not use quotes around the DSN, the prefix set in your TSO profile will be appended to the DSN.

Utility Codes

The following utility codes are valid for the Code field of the Build Utility Procedures screen:

- RO (REORG)
- CK (CHECK INDEX)

- RS (RUNSTATS)
- RC (RECOVER)
- CD (CHECK DATA)
- RT (REPORT TABLESPACE
- CP (COMMAND PROCESSOR)

  Accesses the Utility Command Processor
- IC (IMAGECOPY)
- MC (MERGECOPY)
- ST (STOSPACE)
- MD (MODIFY)
- RR (REPORT RECOVER)
- QU (QUIESCE TABLESPACE)
- MP (MESSAGE PROCESSOR)

  Accesses the message processor.
- PS (CA STATISTICS)

  This option lets you gather statistics for objects in the utility procedure, rather than requiring the execution of an Extract Procedure. This utility code is useful if the selected utilities alter the data. For example, the PS utility code will be useful after a reorganization and image copy (RO, IC, PS).

  Note: For more information about gathering statistics for objects in the utility procedure, see the *CA Database Analyzer User Guide* for more information.
- AM (ACCESS METHODS)

  This option generates IDCAMS JCL. This utility is commonly used for running an ALTER.
- UC (UPDATE CATALOG)

  You can update the DB2 catalog statistics using the most recent CA Database Analyzer statistics if you own CA Database Analyzer.

  Note: For more information about updating the catalog, see the *CA Database Analyzer for DB2 for z/OS User Guide* for more information
- US (USER APPLICATION)
- PR (CA REPORTS)

  This option creates JCL to run CA Database Analyzer report procedures.

  Note: For more information about CA Database Analyzer report procedures, see the *CA Database Analyzer for DB2 for z/OS User Guide*.

Processing

This section discusses the processing options available on the Build Utility Procedures screen:

- Enter or End-Press Enter or End to process the utility request. If you have entered an update request or if undefined symbolic variables are present in the model JCL, you will receive the appropriate screens for entering the required values.
- CANCEL-Use this command to cancel the definition of the utility procedure and return to the Build Utility Extract screen.

View Selected Utilities

You can review the current list of selected utilities and actions by entering the E (explode) Utility Option on the Build Utility Extract panel. The Extract/Action List appear appears.

Note: If you want to view the DB2 objects currently selected, enter the EXPLODE command in the command line.

Screen Layout

An example of the Extract/Action List screen is shown in the following illustration:

Notice the following terms on the previous screen:

SCROLL

Determines the scrolling amount.

PROCEDURE

Identifies the procedure name will always be UTILEXT. UTILEXT is the internal name for the Utility Extract.

O (Option)

Specifies an option next to the Utility procedure or Action Procedure you want to select:

- U-Update the procedure. (In the case of Action Procedures, you are not updating the Action Procedure itself, you are updating a copy of it. Any updates made through this facility will not affect the actual Action Procedure.)
- E-Explode the procedure. Use this option to view the definition.

- When viewing a utility procedure or Action Procedure, you can enter the E option in the utility PARMS column to view the definition of the utilities.
- D-Delete the utility procedure from the Utility Extract. (In the case of Action Procedures, the actual Action Procedure is not deleted.)

PROCNAME

Specifies the name of the Action Procedure or temporary utility procedure.

DESCRIPTION

Specifies the description is always UTILITY EXECUTION.

CREATOR

Specifies the ID of the user who created the procedure.

In the case of a utility procedure, the name will always begin with UTIL and end with a four-digit sequence number. The description will always be the same.

ORD

Specifies the order in which the actions are to be executed. Enter numbers in the order field to number the actions. If you add an action after ordering existing actions, the new action is numbered one higher than the previously highest-number action.

The remaining fields listed for a procedure are described in the CA Database Analyzer User Guide. However, the C and OBJECT DETAIL fields do not apply to Utility Extracts because all procedures are considered unconditional and are tied to the objects selected for the Utility Extract.

Processing

This section describes processing options for the Extract/Action List screen:

- CANCEL or End-Press CANCEL or End to ignore all current options on the screen and return to the Build Utility Extract screen.
- Enter-Press Enter to process options.

Execution

When you are ready to execute your Utility Extract, enter the SUB (SUBMIT) command in the command line of the Build Utility Extract screen and press Enter.

The Submit Utility Job Stream screen appears. For the purpose of this section, the job that is created from a Utility Extract is called a utility job stream.

You have many processing options available on this screen. You can:

- Write the utility job stream to a data set for future review or editing.
- Submit the utility job stream to JES for execution.
- Preview the utility job stream before it is submitted and make any appropriate changes.
- Return to the Build Utility Extract screen to make changes to the Utility Extract.

Model Parameters

Enter the PDS and member name that contains the job card you want to use for the utility job stream. If you are not sure of the member name, you can use standard ISPF member selection criteria for selecting the member from a member list. See Member Selection Lists for more information.

If the model JCL member specified contains symbolic parameters, you will be prompted for their values.

Execution Parameters

Specify the job destination of the utility job stream on the Execution Parameters section of the Submit - Utility Job Stream panel:

- (D)ataset-Writes the utility job stream to a data set.

  You must also enter the destination library and member name for the utility job stream.
- (J)ES-Submits the utility job stream to JES for execution.
- (P)REVIEW-Lets you review the utility job stream before it is submitted to JES or written to a data set.

When you request this option, the Utility Manager writes the utility job stream to a temporary data set in memory. You are then placed into an Edit session on the generated data set. You can view or change the data set as desired.

When you are finished, you are returned to the submit screen. At this point, you can enter the *B* option to submit the reviewed and revised utility job stream to a data set or JES.

Batch Parameters

Two options are available for batch execution. Both require the batch execution mode.

Dataset

Writes the utility job stream to a data set.

You must also enter the destination library and member name for the utility job stream.

JES

Submits the utility job stream to JES for execution.

Member Selection Lists

If you are unsure of the name of the member that contains the *JOB* statement you want to use for the utility job stream, you can request a member selection list. Enter the name of the PDS in the Library field, and enter selection criteria in the Job Card Name field.

Note: You can request a member selection list anywhere a member name is requested, for example on the Symbolic Parameters screen.

An asterisk (*) specifies that all members should be listed. You can limit the number of members listed to those beginning with a certain character string by entering the first characters followed by an asterisk. For example, to list all members that begin with MJ, enter *MJ** in the Job Card Name field. A member list appears:

All model member names begin with MJ as requested. Choose the member you want by entering S next to its name and pressing Enter. If the model JCL member specified contains symbolic parameters, you will be prompted for their values.

Processing

This section describes processing options for the SUBMIT - UTILITY JOB STREAM screen:

- Enter-Press Enter to perform processing based on the Execution Mode:

  If the Execution Mode is set to B, the utility job stream is written to a data set or submitted to JES. In either case, you are returned to the Build Utility Extract screen. You can create a new Utility Extract or End to return to CA RC/Query. The previous Utility Extract is no longer available.

  If the Execution Mode is set to P, you are placed into an Edit session on the utility job stream. When you end the Edit session, you return to the submit screen.

- END or CANCEL-The END or CANCEL command ignores any options on the submit screen. You return to the Build Utility Extract screen where you can make changes to the Utility Extract and resubmit it.

Note: When you submit a Utility Extract (to JES or to a data set), the Utility Extract no longer exists. Therefore, you must END from the submit screen before you submit the utility if you want to change the Utility Extract. For example, if you have previewed the utility job stream and decide you want changes, End from the submit screen, make the changes, and then resubmit the Utility Extract. If you make changes to the utility job stream during the preview session and return to the Build Utility Extract screen, all changes are lost because the utility job stream is regenerated when you request the submit option again.

Job Step Generation

Because many different utilities can be generated in one job stream, the Utility Manager determines the appropriate creation of job steps based on the following rules:

- If the utilities do not require their own set of JCL (the model JCL for the utility contains only control cards), they are combined into one step.

  For example, because RUNSTATS does not require any additional JCL for each invocation of the utility, all RUNSTATS can be included in the same job step. However, an IMAGE COPY does require special JCL (for the output data set name). Therefore, if an IMAGE COPY is requested, each object receives a separate step because each image copy requires its own set of JCL.

- If you are performing multiple DB2 utilities for each DB2 object, all DB2 utilities for the object will be combined into one step. Other non-DB2 utilities (US) will generate their own job step.

  Important! You must ensure that your model JCL does not have conflicting JCL requirements. For example, if two of your model JCL members see the SYSCOPY DD statement, you will receive a JCL error because two SYSCOPY DD statements are present in the job stream. To prevent this error, use a different DD statement for one of the model JCL members.

- You select DB2 utility codes, but some of the utilities refer to model JCL that contains its own JCL statements. In this case, a separate step is generated for each object because there are JCL requirements for each invocation of the utility.

  This kind of step generation occurs when you request a RUNSTATS followed by an IMAGE COPY. The IMAGE COPY contains its own JCL statements.

The following sample cases are presented to help you better understand job step creation. They demonstrate the application of the previous rules.

One Step for All Objects Sample JCL

In this example, you select several DB2 utilities that do not require any additional JCL. All utilities are combined into one step for all objects. The following example shows this type of step creation:

```
//XXXXXX  JOB (XXX,GENERAL),'RCQ',CLASS=A,MSGCLASS=X
//*
//UTIL0001 EXEC PGM=DSNUTILB,REGION=4096K,PARM='PTI'
//STEPLIB  DD  DSN=DSN.CDBALOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//UTPRINT  DD  SYSOUT=*
//SYSUT1  DD  UNIT=SYSDA,SPACE=(CYL,(5,5))
 RUNSTATS TABLESPACE PTDB.PTITSLOG SHRLEVEL REFERENCE
 RUNSTATS TABLESPACE PTDB.PTITSRCE SHRLEVEL REFERENCE
//*
```

One Step for Each Object Based on Special Utility Codes

If you select the same DB2 utilities as discussed, but you also select a non-DB2 utility code (MP, CP, AM, or US). A separate step is generated for *each* object because these special codes require their own job step (they have their own JCL requirements).

In this case, all DB2 utilities for *each* object will be combined into one step followed by the non-DB2 utility steps for the object.

The following example shows this type of step creation:

```
//XXXXXX  JOB (XXX,GENERAL),'RCQ',CLASS=A,MSGCLASS=X
//*
//UTIL0001 EXEC PGM=DSNUTILB,REGION=4096K,PARM='PTI'
//STEPLIB  DD  DSN=DSN.xxxx.CDBALOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//UTPRINT  DD  SYSOUT=*
//SYSUT1  DD  UNIT=SYSDA,SPACE=(CYL,(5,5))
 RUNSTATS TABLESPACE PTDB.PTITSLOG SHRLEVEL REFERENCE
//*
```

```
//*  PTI UTILITY MANAGER
//*
//UTIL0002 EXEC PGM=IKJEFT01
//STEPLIB  DD  DSN=PTIPROD.CDBALOAD,DISP=SHR
//     DD  DSN=DSN.DSNLOAD,DISP=SHR
//PTILIB  DD  DSN=PTIPROD.CDBALOAD,DISP=SHR
//     DD  DSN=DSN.DSNLOAD,DISP=SHR
//PTICLIST DD  DSN=&TEMP1,DISP=(,PASS),UNIT=SYSDA,
// DCB=BLKSIZE=80,SPACE=(TRK,(1,1))
//SYSIN   DD  DSN=&TEMP2,DISP=(,PASS),UNIT=SYSDA,
// DCB=BLKSIZE=80,SPACE=(TRK,(1,1))
//PTIMSG  DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
 CALL 'PTIPROD.CDBALOAD(RALABAT)'
//PTIIPT  DD *
PTI RDPS0200 PTIPH
++CP++PTI
//*
-DISPLAY DATABASE(PTDB)
//UTIL0003 EXEC PGM=DSNUTILB,REGION=4096K,PARM='PTI'
//STEPLIB  DD  DSN=DSN.CDBALOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//UTPRINT  DD  SYSOUT=*
//SYSUT1  DD  UNIT=SYSDA,SPACE=(CYL,(5,5))
 RUNSTATS TABLESPACE PTDB.PTITSRCE SHRLEVEL REFERENCE
//*
//*  PTI UTILITY MANAGER
//*
//UTIL0004 EXEC PGM=IKJEFT01
//STEPLIB  DD  DSN=PTIPROD.CDBALOAD,DISP=SHR
//     DD  DSN=DSN.DSNLOAD,DISP=SHR
//PTILIB  DD  DSN=PTIPROD.CDBALOAD,DISP=SHR
//     DD  DSN=DSN.DSNLOAD,DISP=SHR
//PTICLIST DD  DSN=&TEMP1,DISP=(,PASS),UNIT=SYSDA,
// DCB=BLKSIZE=80,SPACE=(TRK,(1,1))
//SYSIN   DD  DSN=&TEMP2,DISP=(,PASS),UNIT=SYSDA,
// DCB=BLKSIZE=80,SPACE=(TRK,(1,1))
//PTIMSG  DD  SYSOUT=*
```

```
//SYSTSPRT DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD  *
 CALL 'PTIPROD.CDBALOAD(RALABAT)'
//PTIIPT  DD *
PTI RDPS0200 PTIPH

++CP++PTI
-DISPLAY DATABASE(PTDB)
//*
```

One Step for Each Object Based on JCL Requirements

In this example, you select DB2 utility codes, but some of the utilities refer to model JCL that contains its own JCL statements. In this case, a separate step is generated for each object because there are JCL requirements for each invocation of the utility.

This kind of step generation occurs when you request a RUNSTATS followed by an IMAGE COPY. The IMAGE COPY contains its own JCL statements.

The following listing is an example of this type of step creation:

```
//XXXXXX  JOB (XXX,GENERAL),'RCQ',CLASS=A,MSGCLASS=X
//*
//UTIL0001 EXEC PGM=DSNUTILB,REGION=4096K,PARM='ABC'
//STEPLIB  DD  DSN=DSN.CDBALOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//UTPRINT  DD  SYSOUT=*
//SYSUT1  DD  UNIT=SYSDA,SPACE=(CYL,(5,5))
//SYSCOPY1 DD DSN=BACKUP.PTDB.ABCTSLOG.D930202.T113501,
// UNIT=SYSDA,LABEL=(1,SL),DISP=(NEW,DELETE,KEEP)
 RUNSTATS TABLESPACE PTDB.ABCTSLOG SHRLEVEL REFERENCE
 COPY TABLESPACE PTDB.ABCTSLOG DEVT SYSDA COPYDDN
   SYSCOPY1 FULL YES SHRLEVEL REFERENCE
//*
```

```
//UTIL0002 EXEC PGM=DSNUTILB,REGION=4096K,PARM='ABC'
//STEPLIB  DD  DSN=DSN.xxxx.CDBALOAD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//UTPRINT  DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(5,5))
//SYSCOPY1 DD DSN=BACKUP.PTDB.ABCTSRCE.D930202.T113501,
// UNIT=SYSDA,LABEL=(1,SL),DISP=(NEW,DELETE,KEEP)
 RUNSTATS TABLESPACE PTDB.ABCTSRCE SHRLEVEL REFERENCE
 COPY TABLESPACE PTDB.ABCTSRCE DEVT SYSDA COPYDDN
   SYSCOPY1 FULL YES SHRLEVEL REFERENCE
//*
```

Note: If you use the general DB2 utility model, MJUTLGL, the Utility Manager automatically generates a unique step name for each step in the utility job stream. The step name begins with UTIL and ends with a four-digit sequence number (same naming conventions as the temporary utility procedures). The automatic generation of step names requires the use of the %STEPNAME symbolic parameter on your general DB2 utility JCL model.

Restarting Utility Jobs

For information about restarting utility jobs, see the IBM *Command and Utility* Reference.

DB2 Utility IDs

Utility Manager does not assign an explicit DB2 utility ID. Therefore, DB2 uses the userid.jobname as the default utility ID. If you want to assign a specific utility ID, you can change your common utility JCL member (default member MJUTLGL) to add the desired utility ID to the step's EXEC statement. You can use symbolic parameters as part of your specification.

For example, you could specify a utility ID of %USERID.%STEPNAME to automatically generate a unique utility ID for each step in the utility job stream.

Model JCL

Model JCL supplies the Utility Manager with JCL and utility control cards to create the z/OS job stream for executing specified actions in a Utility Extract.

Because each site executing the Utility Manager has different job stream requirements, CA provides model JCL for generating easily customized job streams. The model JCL concept also permits execution of user applications and third-party utilities without your having to furnish any specific requirements.

Model JCL can contain symbolic variables that are automatically replaced by the Utility Manager at execution time or user-defined symbolics that are completed at definition time. For example, the default RUNSTATS model JCL member automatically substitutes the tablespace or indexspace name in the RUNSTATS control cards at job generation time.

When utilities are defined, you must enter PDS member names to use as the model JCL for the specified utilities. Utility Manager provides default model JCL members for each supported utility. You can change the default members as desired. If you select a utility that references model JCL and the model JCL has undefined symbolic parameters, the Symbolic Replacement Variables panel is shown for entering the unknown symbolic variable values.

Note: For more information about using model JCL and the available symbolic variables, see the *CA Database Analyzer for DB2 for z/OS Reference Guide*.

Default Model Members

Utility Manager provides default model JCL members for all supported actions. The default model JCL members are unloaded into a product library at installation time. You can copy these members to your own PDS or reference the install product library. If you want to customize the default models, it is recommended that you copy the members to your own PDS and make the desired changes.

The default model JCL can be customized to meet your specific requirements. You can also create new members to run specific applications or other third party utilities.

You have the option of referencing the INSTALL MODEL library or customizing your own model members. If you customize your model JCL, you will need to copy the MJJOBCD, MJDESTOBJ, MJSTATS, and MJUTLFIL members into your own PDS library, then edit the MJJOBCD member accordingly. If you want to customize many models, it is best to copy all the models to your PDS. This will ensure that a member is not missing, since some of the models may make calls to others.

Some CLIST members that are used for online execution of statistics maintenance are also supplied.

All CLIST members begin with MC. All JCL members begin with MJ. All DB2 utility members begin with MJUTL.

When reviewing the model JCL, you might find the IBM *DB2 Command and Utility Reference* guide helpful. It discusses the syntax and purpose of the DB2 utilities.

Note: For more information about the supplied default members and their descriptions, see the *CA Database Analyzer for DB2 for z/OS Reference Guide*.

Uses

The model JCL contains the following information:

- Job Card-The job card member is specified at execution time. The default model job member is MJJOBCD.
- General DB2 Utility JCL that contains the common JCL statements necessary for executing any DB2 utility-Whenever a DB2 utility is selected, the general DB2 utility JCL is combined with the utility's model JCL to generate the necessary JCL and control cards to execute the utility.

  The general model JCL contains the EXEC statement for executing the IBM DSNUTILB program along with any necessary SYSPRINT, UTPRINT, and other basic JCL requirements needed for executing any DB2 utility.

  Specific JCL requirements such as SYSCOPY DD requirements for an image copy are included in the model JCL member for the image copy option and are not included in the general utility model.

  The default is MJUTLGL.
- Special CA services JCL for executing: Command Processor, Message Processor, and VSAM Access Methods services-These services use member MJUTLPT to supply the necessary JCL.

  Note: You cannot change this member name.
- DB2 Utility JCL for executing supported DB2 utilities-These members specify necessary control cards and any additional JCL requirements (apart from member MJUTLGL) to execute the utility. There is a separate model JCL member for each supported DB2 utility.
- User Applications/Third Party Utilities JCL can use model JCL to specify the necessary job step information for executing the program-You have to create the model JCL members necessary to run the user programs or non-supported utilities. The model JCL can use automatic or user-defined symbolics.

The previous categories are used by the CA Database Analyzer Action Procedure and the Utility Manager facilities. The following categories are used by other CA Database Analyzer Statistics Management facilities:

- Statistics Maintenance JCL or CLIST for executing CA Database Analyzer statistics maintenance in online or batch mode-The default JCL members are MJRTRST and MJBKPST. The default CLIST members are MCRTRST and MCBKPST.
- CA Database Analyzer Procedure Batch Submission JCL for executing CA Database Analyzer procedures in batch mode-The member name must always be MJSTATS, but you can customize the JCL as required. You are prompted for the model JCL library that contains the MJSTATS member at submission time.

Symbolic Variables

To effectively use model JCL, the Utility Manager supports symbolic variables. The two types of symbolic variables are explained in the following section:

- Automatic Symbolic Variables-These are predefined symbolic variables that have a specific meaning to the Utility Manager. For example, they are used to automatically retrieve the name, type, and creator of an object.

  For each object being processed, object-oriented automatic symbolic variables are evaluated. See the following section for a listing of the supported symbolic variables.

- User-Defined Symbolic Variables-You can define your own symbolic variables in the model JCL or CLIST members. You are prompted for the value of user-defined symbolic variables at the time the member is referenced. The Symbolic Replacement Variables screen is shown for entering the values of unknown symbolic variables.

Model JCL Member MJUTLMC Example

Model JCL member MJUTLMC is shown in the following example:

```
//MCUT%INCR DD UNIT=%DUNIT,DISP=(,KEEP)
//  SPACE=(CYL,(%ROSYSUT1,1))
//SYSCOPY DD DSN=BACKUP.%DBNAME.%TSNAME.D%DATE,
// UNIT=%UNIT,LABEL=(1,SL),DISP=(,CATLG)
 MERGECOPY TABLESPACE %DBNAME.%TSNAME %PARTLBL %PART DEVT %UNIT
 NEWCOPY %NEWCOPY WORKDDN MCUT%INCR
```

User-defined Symbolics

The following user-defined symbolics have been specified in MJUTLMC:

%DUNIT

Specifies the device type for the MCUT%INCR work data set.

%UNIT

Specifies the device type for the SYSCOPY data set.

%NEWCOPY

Controls the merging of the image copies. %NEWCOPY is an example of using a user-defined symbolic to control DB2 utility control card parameters.

You are prompted for the values of these symbolics when the Merge Copy utility is executed through a Utility Extract.

Automatic Symbolic Variables

The following automatic symbolics have been defined in MJUTLMC:

%INCR

Generates unique ddnames automatically. If you are executing multiple utilities for each object and the utilities have their own JCL requirements, the Utility Manager generates a separate job step for each DB2 object. If the selected DB2 utilities have conflicting DD statement requirements (for example, if the COPY and MERGECOPY utilities both default to the SYSCOPY DD statement), you need a method of specifying the DD statement name at definition time. Otherwise, you will have conflicting DD statements.

%ROSYSUT1

Allocates primary space, in cylinders, for the MCUT%INCR data set.

%DBNAME/TBNAME

Specifies the selected database/tablespace name.

These symbolics are referenced twice: in the SYSCOPY DD statement and in the control card.

%DATE/TIME

Specifies the date and time as part of the DSN prompt to automatically generate a unique data set name for the work data set.

%PARTLBL/PART

Specifies partition information if a tablespace partition was selected for the Extract Procedure, Utility Manager generates the DSNUM parameters along with the corresponding partition number.

These symbolics are automatically substituted at job stream generation.

Generated JCL

In this example, assume the following:

- Tablespaces ACT and DSN8S71D are selected for the Utility Extract.
- The Merge Copy (MC) utility is selected. The model JCL member used is MJUTLMC. User-defined symbolic variables are entered.
- A merge copy is generated based on the MJUTLMC model.

MJUTLMC can appear as follows:

```
//UTIL0001 EXEC PGM=DSNUTILB,REGION=4096K,PARM='PTI2'

...

//MCUT01 DD UNIT=SYSDA,DISP=(,KEEP),
//  SPACE=(CYL,(1,1))
//SYSCOPY DD DSN=BACKUP.DSN8D71A.ACT.D930202,
// UNIT=SYSDA,LABEL=(1,SL),DISP=(,CATLG)
 MERGECOPY TABLE SPACE DSN8D71A.ACT PART DEVT SYSDA
 NEWCOPY NO WORKDDN MCUT01


...

//UTIL0002 EXEC PGM=DSNUTILB,REGION=4096K,PARM='PTI2'

...
//MCUT02 DD UNIT=SYSDA,DISP=(,KEEP),
//  SPACE=(CYL,(1,1))
//SYSCOPY DD DSN=BACKUP.DSN8D71A.DSN8S71D.D930202,
// UNIT=SYSDA,LABEL=(1,SL),DISP=(,CATLG)
 MERGECOPY TABLE SPACE DSN8D71A.DSN8S71D PART DEVT SYSDA
 NEWCOPY NO WORKDDN MCUT02
```

Model JCL Member MJUTLIC Example

An example of model JCL member MJUTLIC is shown in the following section:

```
//%SYSCPY DD DSN=BACKUP.%DBNAME.%TSNAME.D%DATE.T%TIME,
// UNIT=TAPE,LABEL=(%LBL,SL),DISP=(,CATLG),
// VOL=(,RETAIN%REFER)
 COPY TABLESPACE %DBNAME.%TSNAME DEVT TAPE COPYDDN %SYSCPY
 %PARTLBL %PART FULL %FULL SHRLEVEL REFERENCE
```

User-defined Symbolics

The %FULL user-defined symbolic has been specified in MJUTLIC. %FULL indicates the type of image copy, full or incremental. Yes equals full, and No equals incremental.

Automatic Symbolic Variables

The following automatic symbolics have been defined in MJUTLIC:

%SYSCPY

Specifies the automatically generated ddname.

%LBL

Specifies the LABEL position number.

%REFER

Refers to the volume information specified in the SYSCOPY DD statement (the REFER subparameter of the VOLUME parameter).

%DBNAME/TBNAME

Specifies the selected database/tablespace name.

%DATE/TIME

Specifies the date and time as part of the DSN prompt to automatically generate a unique data set name for the work data set.

%PARTLBL/PART

Specifies partition information if a tablespace partition was selected for the Extract Procedure, Utility Manager generates the DSNUM parameters along with the corresponding partition number.

These symbolics are automatically substituted at job stream generation.

Generated JCL

In this example, assume the following:

- The tablespaces ACT and DSN8S71D are selected for a temporary utility procedure.
- An Image Copy is requested.
- YES is specified for FULL.

The step name lines are included for your reference:

```
//UTIL0001 EXEC PGM=DSNUTILB,REGION=4096K,PARM='PTI2'

...

//SYSCP000 DD DSN=BACKUP.DSN8D71A.ACT.D930322.T101335,
// UNIT=TAPE,LABEL=(001,SL),DISP=(,CATLG),
// VOL=(,RETAIN)
 COPY TABLESPACE DSN8D71A.ACT DEVT TAPE COPYDDN SYSCP000
FULL YES SHRLEVEL REFERENCE
```

```
...

//UTIL0002 EXEC PGM=DSNUTILB,REGION=4096K,PARM='PTI2'

...

//SYSCP000 DD DSN=BACKUP,DSN8D71A.DSN8S71D.D930322.T101335,
// UNIT=TAPE,LABEL=(002,SL),DISP=(,CATLG),
// VOL=(,RETAIN,,REF=*.UTIL0001.SYSCP000)
 COPY TABLESPACE DSN8D71A.DSN8S71D DEVT TAPE COPYDDN SYSCP000
FULL YES SHRLEVEL REFERENCE
```

CALC Example

You can use CALC to calculate the primary and secondary quantities specified when defining VSAM clusters. This example changes the primary and secondary allocations for the underlying VSAM data sets by deleting and defining the clusters. It deletes and redefines a tablespace's VSAM cluster based on the number of pages required for a REORG:

Utility Model

Specifies the general utility model JCL member, MJUTLGL. You will be prompted for the name of the special model JCL member when you specify the US (user application) utility code. In these examples, the special example model is called ALTERJCL, which is a Batch Processor job stream.

CODE

Specifies the utility code US (User Application) is being used.

Before the DELETE CLUSTER statement is executed, the REORG utility will be executed with the UNLOAD PAUSE option. This option specifies that after the data has been unloaded, processing ends. The utility job step is retained so that processing can be restarted in the RELOAD RESTART(PHASE). This can be included as part of the action procedure or as part of the Batch Processor control statement.

DELETE CLUSTER

Deletes the underlying VSAM cluster, deleting all data associated with the tablespace.

DEFINE CLUSTER

Recreates the cluster using the results of the CALC1 variable for primary allocation and CALC2 for secondary allocation. Space will be allocated in tracks.

CALC1 and CALC2

Lists these variables on the Symbolic Replacement Variables panel when you execute the extract. At that time, you will supply the calculations for the primary and secondary quantities.

MODEL

Specifies the model JCL member containing the JCL necessary for the tablespace alter is shown. This model was specified when the Utility Extract prompted you for the user application model JCL.

AUTOMATIC REPLACEMENT

Replaces the %SYSID, %DBNAME, and %TSNAME variables automatically with the DB2 subsystem ID, database name, and tablespace name.

CALC1

Calculates the number of tracks necessary for an IDCAMS DEFINE statement prior to a REORG. (%REORGP) represents the space in pages. This number is increased by 30% (%REORGP*3/10).

Note: The expression is divided by 10 in the example to calculate the tracks for a 3380 drive. The expression should be divided by 12 to calculate the tracks for a 3390 drive.

CALC2

Calculates the secondary quantity to be half of the new primary allocation, in tracks.

The Utility Manager determines the number of pages required for a REORG, makes the calculation, and then substitutes the calculated values for the variables when it creates the DEFINE CLUSTER Batch Processor job:

DEFINE CLUSTER

Specifies the symbolic variables have been automatically replaced with the names of the DB2 subsystem, database, and tablespace.

TRACKS

Specifies the Utility Manager makes the necessary calculations, which results in 10 tracks for the primary allocation, and 5 for the secondary allocation.

After these statements are executed, the REORG utility is restarted by submitting the previous job and specifying RESTART(PHASE).

Customization

You can customize any of the supplied model JCL members to fit your needs. You can add, delete, or modify any of the symbolic variables. You can add completely new members. For example, you might want to have several different members for the Image Copy utility.

You can also change the names of any of the default model JCL members except for MJSTATS and MJUTLPT. *These names must not be* changed.

Message Processor

The Message Processor panel is invoked by the MP utility code. Use this panel to define the destinations of a message and enter the message text.

You can enter up to two lines of text. The text can contain automatic symbolic variables, which are replaced before the message is sent to the designated destinations.

The CA Database Analyzer Action Procedure or Utility Extract will send the message to the specified destinations for each DB2 object selected.

The following fields appear on the Message Processor panel:

Message Destination

Specifies up to three destinations for the messages. You can select any combination of destinations.

CA Log

Indicates whether to send messages to the CA log (Y or N).

Operator Console

Indicates whether to send a message to the operator console. Specify Y, descriptor code, and route code.

Tso User

Indicates whether to send a message to the TSO user. Specify Y and the TSO ID. Symbolic variables are not supported.

Message (Text)

Specifies the message text, up to two lines. The message text can contain automatic symbolic variables. User-defined symbolics are ignored.

Press Enter or END to save the message parameters and return to the build panel. If there are errors, you must correct them before you can exit.

If you specify CANCEL, the message definition is cancelled. If the message is undefined due to the cancel, a U (update) is shown under the PARMS column on the build screen. If you want to save the procedure, you must correct the utility parameters or erase the utility code.

Utility Command Processor

The DB2 Command Request panel lets you enter up to 10 DB2 commands. If you need to enter more, you can re-enter the CP utility code. The CP utility code always generates its own utility step. It is *not* combined with any other DB2 utility codes. However, the CP utilities are combined with any MP (Message Processor) or AM (VSAM) utility statements. The same program processes these special utility requests and they are all combined into the same job step. A separate step is created for *each* processed DB2 object.

Utility Manager does not check the command for valid syntax. The DB2 commands are sent to the DSN processor as entered.

You cannot specify user-defined symbolic variables. However, automatic symbolic variables can be specified as part of the DB2 commands. For example, you can enter the following command:

```
-DISPLAY DATABASE(%DBNAME)
```

DB2 commands are executed for *each* object selected by the CA Database Analyzer action procedure or utility extract.

For a listing of possible DB2 commands, see the IBM *DB2 Command and Utility Reference* guide. This guide does not cover DB2 command syntax or usage.

Utility Manager invokes the TSO DSN command processor to process the DB2 commands.

Note: The Utility Processor is accessed by specifying the CP utility code. The Utility Command Processor is for attaching DB2 commands to your utility extract. Do not confuse it with the *interactive* Command Processor facility accessed by entering the DB2C command.

Press Enter or END to save the specified commands and return to the Utility Procedures Build panel.

Action Procedures

CA Database Analyzer action procedures are definitions of utilities to be executed under specific conditions. They can be defined as conditional (execute when specific thresholds are reached) or unconditional (always executed).

Note: You must have CA Database Analyzer installed to create and use Action Procedures.

There are many instances when the utilities you want to execute against a selected set of objects have already been defined as a CA Database Analyzer Action Procedure. The Utility Manager lets you use predefined Action Procedures as part of your Utility Extract.

When Action Procedures are selected for a Utility Extract, they are always treated as unconditional.

You can combine Action Procedures with utility procedures in the same Utility Extract. You can create any combination of Action Procedures and utility procedures as necessary to perform the requested processing.

Action Procedure Services

Action Procedure Services permit you to use existing Action Procedures for your Utility Extract.

To select Action Procedures for a Utility Extract, enter the *S* (select) Utility Option on the Build Utility Extract screen.

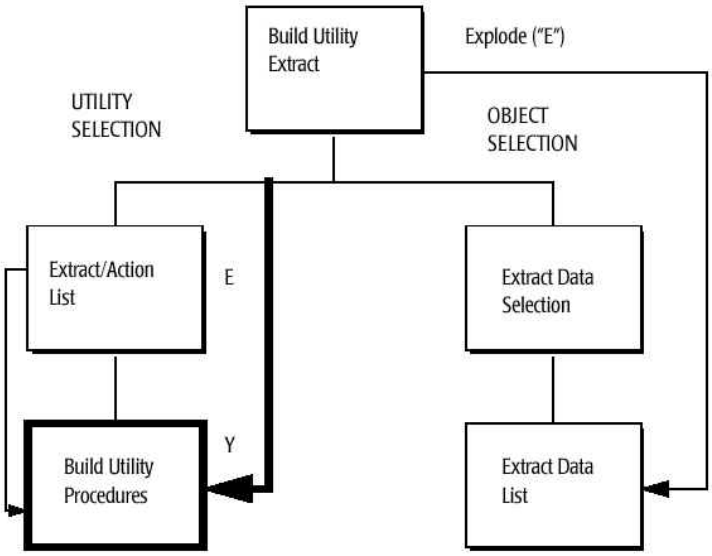
The Action Procedure Services screen is shown, listing all available Action Procedures. Enter **S** next to an Action Procedures to select it for the Utility Extract. Enter **E** next to an Action Procedure to view its current definition.

All Action Procedures are considered unconditional when they are used by a Utility Extract.

When the Action Procedure Services screen is invoked by the Utility Manager, it does not support the delete or update options. To perform these operations, you must use the Action Procedures option on the CA Database Analyzer Main Menu.

Flow Chart

If you own CA Database Analyzer, the possible Utility Manager processing flow is slightly different than the one shown in the following illustration. If you own CA Database Analyzer, you can also access Action Procedure Services from the Utility Manager:



The Action Procedure Services component is shown in bold lines.

For more information about Action Procedure services, see the *CA Database Analyzer User Guide*.

Index

E

F

G

H

I

T

U

V

Y