CATALOG MANAGER for DB2®
User Guide

Supporting

Version 10.1 of Administrative Assistant for DB2
Version 10.1 of CATALOG MANAGER for DB2
Version 10.1 of Database Administration for DB2
Version 10.1 of System Performance for DB2

April 2011
Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address
BMC SOFTWARE INC
2101 CITYWEST BLVD
HOUSTON TX 77042-2827
USA

Telephone
713 918 8800 or
800 841 2031

Fax
713 918 8000

Outside United States and Canada

Telephone
(01) 713 918 8800

Fax
(01) 713 918 8000

© Copyright 1998-2011 BMC Software, Inc.

BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. All other trademarks or registered trademarks are the property of their respective owners.

DB2, DFSMS, DFSMSshm, IBM, MVS, OS/390, QMF, RACF, REXX, z/OS, and zSeries are the trademarks or registered trademarks of International Business Machines Corporation.

Java is a registered trademark of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

The information included in this documentation is the proprietary and confidential information of BMC Software, Inc., its affiliates, or licensors. Your use of this information is subject to the terms and conditions of the applicable End User License agreement for the product and to the proprietary and restricted rights notices included in the product documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”

Support website

You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can

- read overviews about support services and programs that BMC offers
- find the most current information about BMC products
- search a database for issues similar to yours and possible solutions
- order or download product documentation
- download products and maintenance
- report an issue or ask a question
- subscribe to receive proactive e-mail alerts when new product notices are released
- find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter SupID:<yourSupportContractID>, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC

Have the following information available so that Customer Support can begin working on your issue immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the issue
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as file system full
  - messages from related software
License key and password information

If you have questions about your license key or password, use one of the following methods to get assistance:

- Send an e-mail message to customer_support@bmc.com.
## Contents

### About this book
- Related publications ................................. 20
- Conventions ........................................ 20
- Syntax statements .................................. 21

### Chapter 1  Introducing CATALOG MANAGER 23
- Introduction ........................................ 23
- CATALOG MANAGER functions .................. 24
  - Listing DB2 catalog objects ......................... 25
  - Creating objects .................................. 25
  - Searching the DB2 catalog ......................... 25
  - Describing objects ................................ 25
  - Manipulating data .................................. 26
  - Executing commands and statements .............. 26
  - Using utility profiles .............................. 26
  - Recovering dropped objects ....................... 27
  - Maintaining action logs ......................... 27
  - Accessing a different DB2 subsystem ............. 28
  - Updating and executing SQL ....................... 28
  - Managing authorizations ......................... 29
  - Using an indirect catalog ......................... 29
  - Running SQL Explorer ............................ 29
- Integration with BMC solutions ................... 30
  - Administrative Assistant for DB2 ............... 30
  - Database Administration for DB2 ............... 30
  - System Performance for DB2 ..................... 30
- Where to go from here .............................. 31

### Chapter 2  Getting started with CATALOG MANAGER 33
- Before you begin .................................... 34
- Accessing CATALOG MANAGER functions ........ 34
- Using the Primary Menu panel ..................... 36
  - Using the Command line ........................... 37
  - Using a command recognition character .......... 37
  - Selecting an Action ............................... 38
  - Identifying an object type ....................... 38
  - Specifying a qualifier ............................ 42
- Using nonprintable or nonviewable characters .... 44
Using ISPF commands .................................................. 45
Getting help .............................................................. 46
Navigation in CATALOG MANAGER ................................ 47
Generating lists in CATALOG MANAGER ......................... 47
Using a list panel ......................................................... 48
Generating an object list from the Primary Menu panel ........ 49
Generating secondary lists ............................................. 50
Listing and executing commands ..................................... 56
Displaying the online command reference ....................... 56
Issuing command prefixes ............................................. 58
Abbreviating command keywords and object names .......... 58
Omitting object types from commands ............................. 58
Including multiple objects in a command ......................... 59
Issuing commands from the Cmd column ......................... 59
Issuing Wait-for-Enter commands against multiple objects 59
Using Fast Path Navigation ........................................... 63
Where to go from here .................................................. 64

Chapter 3 Setting up CATALOG MANAGER .......................... 65
Before you begin ......................................................... 66
Viewing settings ......................................................... 66
Viewing information about the environment ..................... 67
Viewing DB2 initialization parameters ............................. 67
Viewing the CATALOG MANAGER common area .............. 67
Viewing and updating DB2 special registers ..................... 67
Granting access to CATALOG MANAGER functions ............ 69
Using Execution plans .................................................. 70
Manipulating plans ...................................................... 71
Setting the MEMLIMIT system parameter ....................... 71
Using options to control your environment ...................... 71
Using the installation options ........................................ 72
Using the user options ................................................ 72
Using the product options ............................................ 73
Putting it all together .................................................. 75
Setting user options .................................................... 76
Defining an options data set ........................................... 77
Setting basic options .................................................. 79
Setting general options ............................................... 82
Setting object use options ............................................ 85
Setting JCL Generation options ..................................... 86
Setting data set options ............................................... 88
Setting SQL and confirm options ................................... 90
Setting SQL SELECT options ......................................... 93
Setting panel graphic options ....................................... 94
Setting CATALOG MANAGER switches ......................... 95
Setting DESCRIBE options .......................................... 98
Setting product options ............................................... 99
Setting the JCL options for job cards .............................. 101
Setting the JCL options for STEPLIBs ............................. 104
| Setting the JCL options for static data sets | 105 |
| Setting the JCL options for tapes | 109 |
| Setting the JCL options for temporary work data sets | 112 |
| Setting the JCL options for permanent data sets | 114 |
| Setting the JCL generation data group options | 120 |
| Setting the JCL debugging, display, and Execution options | 121 |
| Setting the JCL utility installation options module name options | 124 |
| Setting the online reorg options | 126 |
| Setting the non-worklist JCL options | 127 |
| Setting the LISTDEF and TEMPLATE data set options | 129 |
| Setting user variables | 130 |
| Creating a user POF | 132 |
| Updating a user POF | 133 |
| Using multiple POFs | 134 |
| Refreshing the initial POF | 135 |
| Generating POF reports | 136 |
| Reusing a POF in a subsequent installation | 137 |
| Overriding POF values in SLIBs | 138 |
| Adding steps to the JCL | 140 |
| Obtaining a list of TEMPLATEs or LISTDEFS in CATALOG MANAGER | 143 |
| Using the commands table | 143 |
| Commands table syntax and parameters | 145 |
| Creating a user commands table | 148 |
| Retaining a user commands table when upgrading CATALOG MANAGER | 151 |
| Writing user commands as CLISTs | 152 |
| Development aids for user commands | 152 |
| Command program parameters | 153 |
| Passing object type and name | 154 |
| CATALOG MANAGER tables | 156 |
| Where to go from here | 157 |

**Chapter 4  Accessing other DB2 subsystems**

| Overview | 159 |
| Using the DB2 Attach feature | 160 |
| Attaching CATALOG MANAGER to a specified SSID | 161 |
| Attaching to an SSID or server by using the connection selection list | 162 |
| Switching catalog access | 163 |
| Restoring the default attachment | 164 |
| Using the DB2 Connect feature | 164 |
| Connecting to a specified SSID | 166 |
| Obtaining data set information from a remote SSID | 168 |
| Connecting to a remote SSID from a location list | 169 |
| Connecting to a remote SSID and issuing DB2 commands to display, start, or stop objects | 169 |
| Using saved connections | 170 |
| Viewing the connections table | 173 |
| Using DB2-identifiers with the CONNECT command | 174 |
| Identifying attachments or connections with unique values | 175 |
| Displaying the current server ID | 175 |
# CATALOG MANAGER for DB2 User Guide

## Chapter 5 Working with lists and searches

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>178</td>
</tr>
<tr>
<td>Using mixed lists</td>
<td>178</td>
</tr>
<tr>
<td>Valid source objects for mixed lists</td>
<td>178</td>
</tr>
<tr>
<td>Generating a mixed list</td>
<td>179</td>
</tr>
<tr>
<td>Using combined lists</td>
<td>180</td>
</tr>
<tr>
<td>Generating a combined list</td>
<td>181</td>
</tr>
<tr>
<td>Excluding objects from a combined list</td>
<td>182</td>
</tr>
<tr>
<td>Using SEARCH to generate lists based on object attributes</td>
<td>183</td>
</tr>
<tr>
<td>Valid objects for searches</td>
<td>183</td>
</tr>
<tr>
<td>Generating a list by using the SEARCH command</td>
<td>185</td>
</tr>
<tr>
<td>Using host variables in a search</td>
<td>190</td>
</tr>
<tr>
<td>Using the Quick-Search feature</td>
<td>192</td>
</tr>
<tr>
<td>Using saved search variables in a Quick-Search</td>
<td>193</td>
</tr>
<tr>
<td>Using a WHERE clause in a Quick-Search</td>
<td>194</td>
</tr>
<tr>
<td>Including Quick-Searchs in BATCH jobs</td>
<td>194</td>
</tr>
<tr>
<td>Creating complex searches</td>
<td>194</td>
</tr>
<tr>
<td>Creating searches that do not contain a JOIN</td>
<td>195</td>
</tr>
<tr>
<td>Creating searches that contain a JOIN</td>
<td>195</td>
</tr>
<tr>
<td>Customizing object list displays</td>
<td>196</td>
</tr>
<tr>
<td>Specifying a new order for displayed columns</td>
<td>197</td>
</tr>
<tr>
<td>Sorting a list by one or more columns</td>
<td>198</td>
</tr>
<tr>
<td>Locating string values</td>
<td>200</td>
</tr>
<tr>
<td>Counting items</td>
<td>200</td>
</tr>
<tr>
<td>Describing list objects</td>
<td>201</td>
</tr>
<tr>
<td>Generating JCL for a job in batch</td>
<td>206</td>
</tr>
<tr>
<td>Using the BATCH command for a DB2 object list or a mixed list</td>
<td>207</td>
</tr>
<tr>
<td>Using the BATCH command for a CATALOG MANAGER list or search</td>
<td>210</td>
</tr>
<tr>
<td>Generating, editing, and executing SQL</td>
<td>212</td>
</tr>
<tr>
<td>Using Confirm SQL panels</td>
<td>212</td>
</tr>
<tr>
<td>Working with the SQL_Table</td>
<td>213</td>
</tr>
<tr>
<td>Applying SQL model statements</td>
<td>215</td>
</tr>
<tr>
<td>Extended SQL processing</td>
<td>219</td>
</tr>
<tr>
<td>Where to go from here</td>
<td>222</td>
</tr>
</tbody>
</table>

## Chapter 6 Browsing and editing data

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>223</td>
</tr>
<tr>
<td>Browsing table data</td>
<td>224</td>
</tr>
<tr>
<td>Invoking the data browsing function</td>
<td>224</td>
</tr>
<tr>
<td>Setting options for browsing data</td>
<td>224</td>
</tr>
<tr>
<td>Browsing data</td>
<td>229</td>
</tr>
<tr>
<td>Browsing data in LOB columns</td>
<td>230</td>
</tr>
</tbody>
</table>

---

Troubleshooting a DB2 Attach or DB2 Connect failure: 176
Where to go from here: 176
Figures

Example of BMC Software Administrative Products for DB2 panel .................. 34
CATALOG MANAGER Primary Menu panel .................................................. 36
Navigating through CATALOG MANAGER .................................................. 47
Object list ................................................................................................. 48
Generating a table space list from a database list ........................................... 51
Table Space List panel ............................................................................... 51
Generating table lists from multiple table spaces ........................................... 52
Table List panel for first source table space .................................................. 53
Table List panel for next source table space ................................................. 53
Generating lists of varied dependent objects ................................................. 54
Index List panel for first source table .......................................................... 55
Column List panel for second source table ................................................... 55
Scrollable Commands List panel ............................................................... 57
Tablespace List panel with line commands for copying multiple objects ........ 60
Tablespace List panel with shortcut commands for copying multiple table spaces ........................................................................................................... 61
Tablespace List panel with command to copy all listed table spaces ............... 62
Tablespace List panel with exclude commands ............................................. 62
DB2 Special Registers panel ....................................................................... 68
How CATALOG MANAGER uses options ................................................... 75
Options Dataset not defined panel .............................................................. 78
Allocate Data Set ....................................................................................... 79
CATALOG MANAGER Options panel ......................................................... 80
Options Dataset has changed panel ............................................................. 81
General Options panel .............................................................................. 83
Object Use Options panel .......................................................................... 85
JCL Generation Options panel ................................................................... 87
Datasets panel ........................................................................................... 89
SQL and Confirm Options panel ............................................................... 91
SQL Select panel ....................................................................................... 93
Colors panel ............................................................................................ 95
Switches panel ......................................................................................... 96
Describe Options panel ............................................................................ 99
JCL Generation Update Panel .................................................................... 100
JCL Generation Jobcard Options Update panel ........................................... 102
JCL Generation STEPLIB Options Update panel ........................................ 104
JCL Generation Static Data Set Options panel ............................................ 106
JCL Generation Tape Options Update panel ............................................... 110
JCL Generation Options For Sort Files Update panel .................................... 113
JCL Generation Data Set Options For Sortout Update panel ......................... 116
Figure 15
Tables

Description of Primary Menu panel .................................................. 36
Object types displayed on the Primary Menu panel ............................. 38
Object types not displayed on the Primary Menu panel ....................... 39
Four-character object types .......................................................... 42
Wildcard characters supported by CATALOG MANAGER ...................... 43
Wildcards in fixed-length CHAR columns ........................................ 43
How CATALOG MANAGER qualifies two-part object names ................... 44
Qualifier exceptions ................................................................. 44
Nonprintable or nonviewable characters ............................................ 45
ISPF commands ....................................................................... 45
Fast Path Navigation Commands ...................................................... 63
CATALOG MANAGER plans .......................................................... 69
Options panels ..................................................................... 76
Valid values for the Default SQLID field .......................................... 82
Valid values for the SET CURRENT SQLID to field ............................ 85
Data set sizing options ................................................................. 107
Work data sets in the JCL cleanup job step ........................................ 108
Work data sets used by utilities ......................................................... 109
Values for 7-track tape drives ........................................................ 111
Comment codes for data set sizing ................................................... 123
Syntax of commands table entry ....................................................... 145
Commands table variables ................................................................ 145
Customizable code for building user commands .................................... 152
Key parameters for a user command program .................................... 153
Parameters for use with the $ACTULOG macro .................................. 154
Passing object types and names in user-written commands ............... 155
ISPF variables for user commands or CLISTs .................................... 156
CATALOG MANAGER tables ......................................................... 156
CONNECT command parameters .................................................... 167
Columns on the Connections Table panel ......................................... 174
Valid source objects for mixed lists .................................................. 178
Valid source objects for generating combined lists ............................... 181
Valid source objects for searches ..................................................... 183
Valid search operators ................................................................... 186
Host variable values .................................................................... 192
DES command descriptions ............................................................. 204
DB2 command syntax .................................................................. 210
Commands to invoke data browsing ................................................ 224
SELECT statement specifications ..................................................... 226
SELECT statement specification panel commands ............................. 227
About this book

This book contains detailed information about the BMC Software CATALOG MANAGER for DB2 product for the DB2 DBMS. It is intended for system administrators and database administrators (DBAs).

To use this book, you should be familiar with the following items:

- IBM® DB2 Universal Database for z/OS® and OS/390®
- Interactive System Productivity Facility (ISPF)

Like most BMC documentation, this book is available in printed and online formats. To request printed books or to view online books and notices (such as release notes and technical bulletins), see the Customer Support website at http://www.bmc.com/support. Most product shipments also include the books on a documentation CD.

**NOTE**

Online books are formatted as PDF or HTML files. To view, print, or copy PDF books, use the free Adobe Reader from Adobe Systems. If your product installation does not install the reader, you can obtain the reader at http://www.adobe.com.

The software also offers online Help. To access Help, press F1 within any product or click the Help button in graphical user interfaces (GUIs).

To access Help for a panel, press HELP. Some panels also offer field-level Help. To access field-level Help, place the cursor on a field, and then press HELP.
Related publications

The following related publications supplement this book and the online Help:

<table>
<thead>
<tr>
<th>Category</th>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>- Administrative Products for DB2 Installation Guide</td>
<td>contains information about installing CATALOG MANAGER</td>
</tr>
<tr>
<td></td>
<td>- Administrative Assistant for DB2 Installation Guide</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- BMC Products and Solutions for DB2 for zOS Installation Planning Guide</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Database Administration for DB2 Installation Guide</td>
<td></td>
</tr>
<tr>
<td>Messages</td>
<td>Administrative Products for DB2 Messages Manual</td>
<td>contains descriptions of and responses for the information, warning, and error messages that CATALOG MANAGER generates</td>
</tr>
<tr>
<td>Use of associated product</td>
<td>SQL Explorer for DB2 User Guide</td>
<td>includes instructions for using the SQL Explorer for DB2 product to perform SQL tuning and analysis</td>
</tr>
<tr>
<td>Notices</td>
<td>release notes, flashes, and technical bulletins</td>
<td>contain timely information about CATALOG MANAGER, such as revisions to the installation instructions and updated product information between product releases</td>
</tr>
</tbody>
</table>

Conventions

This book uses the following special conventions:

- **All syntax, operating system terms, and literal examples are presented in this typeface.**

- **Variable text in path names, system messages, or syntax is displayed in *italic* text:**

  `testsys/instance/fileName`

- The symbol `=>` connects items in a menu sequence. For example, **Actions => Create Test** instructs you to choose the **Create Test** command from the **Actions** menu.
Syntax statements

The following example shows a sample syntax statement:

```
COMMAND KEYWORD1 [KEYWORD2 | KEYWORD3] KEYWORD4={YES | NO} fileName...
```

The following table explains conventions for syntax statements and provides examples:

<table>
<thead>
<tr>
<th>Item</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items in italic type represent variables that you must replace with</td>
<td>alias</td>
</tr>
<tr>
<td>a name or value. If a variable is represented by two or more</td>
<td>databaseDirectory</td>
</tr>
<tr>
<td>words, initial capitals distinguish the second and subsequent words.</td>
<td>serverHostName</td>
</tr>
<tr>
<td>Brackets indicate a group of optional items. Do not type the</td>
<td>[tableName, columnName, field]</td>
</tr>
<tr>
<td>brackets when you enter the option. A comma means that you can choose</td>
<td>[-full, -incremental, -level] (UNIX)</td>
</tr>
<tr>
<td>one or more of the listed options. You must use a comma to separate</td>
<td></td>
</tr>
<tr>
<td>the options if you choose more than one option.</td>
<td></td>
</tr>
<tr>
<td>Braces indicate that at least one of the enclosed items is required.</td>
<td>(DBDName</td>
</tr>
<tr>
<td>Do not type the braces when you enter the item.</td>
<td>UNLOAD device={disk</td>
</tr>
<tr>
<td>A vertical bar means that you can choose only one of the listed</td>
<td>(-a</td>
</tr>
<tr>
<td>items. In the example, you would choose either commit or cancel.</td>
<td>{commit</td>
</tr>
<tr>
<td>An ellipsis indicates that you can repeat the previous item or items</td>
<td>{commit</td>
</tr>
<tr>
<td>as many times as necessary.</td>
<td>columnName . . .</td>
</tr>
</tbody>
</table>
Introducing CATALOG MANAGER

This chapter contains the following topics:

Introduction ......................................................... 23
CATALOG MANAGER functions ........................................ 24
  Listing DB2 catalog objects ........................................ 25
  Creating objects .................................................. 25
  Searching the DB2 catalog ....................................... 25
  Describing objects ................................................ 25
  Manipulating data ............................................... 26
  Executing commands and statements ............................. 26
  Using utility profiles ............................................. 26
  Recovering dropped objects ...................................... 27
  Maintaining action logs ........................................ 27
  Accessing a different DB2 subsystem ............................ 28
  Updating and executing SQL ..................................... 28
  Managing authorizations ....................................... 29
  Using an indirect catalog ....................................... 29
  Running SQL Explorer .......................................... 29
Integration with BMC solutions .................................... 30
  Administrative Assistant for DB2 ................................. 30
  Database Administration for DB2 ................................. 30
  System Performance for DB2 .................................. 30
Where to go from here ............................................ 31

Introduction

The BMC CATALOG MANAGER for DB2 product provides an efficient gateway to DB2 catalog information. CATALOG MANAGER features highly productive methods for creating and managing your DB2 databases. Using an interface based on the Interactive System Productivity Facility (ISPF), CATALOG MANAGER provides interactive access to catalog information and application data with simple-to-use menus, panels, and online Help.
Using CATALOG MANAGER, you interact with the catalog by performing actions on specific objects. You do not need to have complete knowledge of DB2 structures or Structured Query Language (SQL) syntax because CATALOG MANAGER maintains database structures and constructs the necessary SQL statements. You choose when and how to execute these statements. The created SQL can be saved, edited, and reused for other tasks.

CATALOG MANAGER facilitates information retrieval from the DB2 catalog by producing lists. The primary productivity benefit comes from your ability to initiate an action on an object, directly from a list. For example, you can issue a command on an object that will generate an SQL statement for the object. CATALOG MANAGER generates the statement based on your specifications and passes the statement to DB2 to validate the SQL statement, execute the statement, and provide SQL return codes (SQLCODEs). Other products in the BMC family of Administrative products, such as the BMC ALTER for DB2 and BMC CHANGE MANAGER for DB2 products, implement changes that require objects to be dropped and re-created.

CATALOG MANAGER functions

CATALOG MANAGER provides the following functions:

- listing DB2 catalog objects
- creating objects
- searching the DB2 catalog
- describing objects
- manipulating data
- executing commands
- using utility profiles
- recovering dropped objects
- maintaining action logs
- accessing different DB2 subsystems
- saving, editing, and executing SQL
- managing authorizations
- using an indirect catalog
- running SQL Explorer
Listing DB2 catalog objects

One of the most useful functions of CATALOG MANAGER is its ability to generate lists of DB2 catalog objects, both for queries and for executing commands against the listed items. The ability to execute action commands against list items offers powerful administrative support in the DB2 production environment.

The LIST function of CATALOG MANAGER enables you to work with the DB2 catalog easily. You generate object lists by using actions and commands with the object type code. From those object lists, you can often execute commands or actions to generate more object lists.

Each list includes all of the columns of the catalog table for the object type. Besides providing a source of useful catalog information, lists can be sorted and their objects can be manipulated. You can print any list, and you can sort any list on any column, in ascending or descending order.

Creating objects

CATALOG MANAGER lets you create entire hierarchies by generating the SQL that is required to create objects and their hierarchical dependents. You can reference an object to create a new object that has similar attributes. This method saves time by providing almost all of the necessary SQL for the CREATE statement.

Searching the DB2 catalog

CATALOG MANAGER enables you to search DB2 catalogs by using descriptive criteria to generate a list of very specific objects. You can also save the search criteria, which enables you to repeat the search quickly and easily.

Describing objects

CATALOG MANAGER enables you to generate a description of any object in a list. The detailed description contains relevant information that is stored in the DB2 catalog about a specific object, including structure, statistics, and dependencies.
Manipulating data

You can manipulate data in the following ways:

- You can use the data editing and data browsing functions to create, edit, or browse data in a table or view without leaving CATALOG MANAGER. You can also copy data from one table or view into another table or view. In many cases, this feature prevents you from having to run load and unload utilities.

- You can use the data manipulation language (DML) statements (such as SELECT, INSERT, UPDATE, and DELETE) that CATALOG MANAGER automatically builds for you.

Executing commands and statements

In addition to using CATALOG MANAGER as an information tool, you can execute SQL statements and DB2 commands and submit utility jobs interactively. The following types of commands and statements are processed through CATALOG MANAGER:

- SQL statements
- DB2, DB2 DSN, and DB2 utility commands
- BMC utility commands
- BMCSTATS and BMC SQL Explorer for DB2 commands
- user-written commands

The execution of commands and statements with CATALOG MANAGER requires minimum input—usually one command verb. You do not need to know the syntax of the eventual SQL, command, or utility statement because CATALOG MANAGER constructs the required DB2 syntax from information in the selected line of a list, installation defaults, and user-specific defaults. Where appropriate, you can modify the options that are used.

Using utility profiles

A utility profile is a file that contains customized specifications for the syntax of each type of utility job that you generate. Using utility profiles saves you time and helps avoid user errors because you avoid having to specify the syntax each time that you generate a job.
CATALOG MANAGER creates the following types of utility profiles:

- site profiles, which apply to all users
- user profiles, which are available to the users who create them

**Recovering dropped objects**

CATALOG MANAGER provides options and features that can show you the results of dropping objects and help you recover dropped object structures and data:

- To help you manage dropping and recovering more efficiently, you can display a list of dependent objects that will be dropped when the current DROP statement is executed.
- You can use data definition language (DDL) to re-create structures.
- You can use the last full image copy to recover data.

**Maintaining action logs**

CATALOG MANAGER maintains the following logs of user actions:

- The *Session Log* shows who executed a particular CATALOG MANAGER command and when the command was executed. The log also indicates other activity by that user during the same session. As an aid to maintaining system integrity, you can use CATALOG MANAGER to track every occurrence of any or all commands that are executed in CATALOG MANAGER.

- The *DDL Audit Log* contains an entry for any execution of an SQL statement or a command using SQL that might change the DB2 catalog (for example, `COPYAUTHS` and `DROPRECOVERY`).

- The *Drop Recovery Log* contains each SQL statement that is necessary to recover the catalog structure of a dropped object and its dependencies.

You can display these action logs. As with object lists, you can display the whole log or a partial log based on specific search criteria. If you are authorized to do so, you can also purge log entries that are older than a specified date.
Accessing a different DB2 subsystem

CATALOG MANAGER provides two methods of accessing other DB2 subsystems:

- **DB2 Attach**

  The DB2 Attach feature enables you to change the CATALOG MANAGER session that you are running without leaving CATALOG MANAGER. You simply disconnect from the current SSID and attach to a different SSID through the Call Attach Facility (CAF).

- **DB2 Connect**

  If DB2 Distributed Data Facility (DDF) is installed, the DB2 Connect feature enables you to connect to a DB2 subsystem in another z/OS system without terminating your current CATALOG MANAGER session. Through this connection, you can run SQL.

Updating and executing SQL

CATALOG MANAGER provides simple methods for editing, saving, and executing the SQL that you generate from menu selections and commands. Panels in CATALOG MANAGER display the SQL statements that CATALOG MANAGER generates and allow you to specify the default settings of options that SQL commands use. For example, you can

- name and save the SQL for reuse, edit the SQL, and execute the SQL
- display a list of all of the saved SQL statements and then execute any or all of them

Both of these features help you generate and execute SQL quickly and easily.

NOTE

Other BMC programs can view and update the DDL Audit Log and the Drop Recovery Log.
Managing authorizations

Managing authorizations within DB2 can be one of the most complex tasks that a DBA or system administrator must perform. CATALOG MANAGER lets you generate a list of users or objects and see the privileges that have been granted to those users or objects. From this list, you can perform the following actions:

- grant and revoke authorizations on catalog objects
- reassign authorizations
- copy established authorizations from one user to another user, or from one object to another object

Accessing DB2 catalog tables requires the appropriate DB2 user authorization. Using CATALOG MANAGER to access information that is stored in the DB2 catalog tables does not increase or change your authorization level for DB2.

Using an indirect catalog

CATALOG MANAGER provides a procedure for using a copy or view of the catalog for many commands, avoiding catalog contention among several users. This procedure is called catalog indirection.

During installation, you can create copies or views, and then assign synonyms to them. All of the CATALOG MANAGER information commands then use these synonyms to access the copy or view. Commands that update the catalog, however, execute against the actual catalog.

Running SQL Explorer

The BMC SQL Explorer product is an SQL analysis tool that enables you to solve performance problems that result from inefficient SQL statements. You can use CATALOG MANAGER commands to navigate to SQL Explorer and perform the following tasks:

- access SQL
- explain DBRMs, packages, and plans
- explain individual SQL statements in DBRMs and packages
Integration with BMC solutions

CATALOG MANAGER is a component of the following BMC solutions:

- BMC Administrative Assistant for DB2
- BMC Database Administration for DB2

In addition, the Catalog Browse functionality of CATALOG MANAGER is a component of the BMC System Performance for DB2 solution.

Administrative Assistant for DB2

The Administrative Assistant solution enables users of all experience levels to navigate through the DB2 catalog quickly and to manage a complex DB2 environment easily. For more information, see the Administrative Assistant for DB2 Release Notes and the Administrative Assistant for DB2 Installation Guide.

Database Administration for DB2

You can use the Database Administration solution to manage your DB2 databases quickly, efficiently, and effectively. For more information, see the Database Administration for DB2 Release Notes and the Database Administration for DB2 Installation Guide.

System Performance for DB2

The BMC System Performance for DB2 solution combines the features and functionality of a number of components to help you increase staff productivity and maintain performance consistency by tuning your DB2 system dynamically and automatically as workloads change. For more information, see the System Performance for DB2 Release Notes and the System and SQL Performance for DB2 Installation Guide.
Where to go from here

Now that you know about the features of CATALOG MANAGER, you are ready to start using them to enhance your productivity. Chapter 2, “Getting started with CATALOG MANAGER,” provides the information that you need to understand the design of CATALOG MANAGER and to use its most basic features.
Chapter 2 Getting started with CATALOG MANAGER

This chapter contains the following topics:

Before you begin .................................................. 34
Accessing CATALOG MANAGER functions .................. 34
Using the Primary Menu panel .................................. 36
    Using the Command line ..................................... 37
    Using a command recognition character .................. 37
    Selecting an Action .......................................... 38
    Identifying an object type .................................. 38
    Specifying a qualifier ...................................... 42
Using nonprintable or nonviewable characters ................ 44
Using ISPF commands .......................................... 45
Getting help ....................................................... 46
Navigation in CATALOG MANAGER .......................... 47
Generating lists in CATALOG MANAGER .................... 47
    Using a list panel ............................................ 48
    Generating an object list from the Primary Menu panel .. 49
    Generating secondary lists .................................. 50
Listing and executing commands .............................. 56
    Displaying the online command reference ................. 56
    Issuing command prefixes .................................. 58
    Abbreviating command keywords and object names ........ 58
    Omitting object types from commands .................... 58
    Including multiple objects in a command ................. 59
    Issuing commands from the Cmd column .................. 59
    Issuing Wait-for-Enter commands against multiple objects 59
Using Fast Path Navigation .................................... 63
Where to go from here .......................................... 64
Before you begin

Before you use the CATALOG MANAGER for DB2 product, ensure that you have completed all of the required installation and customization procedures. For more information, see the installation guide.

In addition, in your ISPF settings, remove the `/` for the Long message in pop-up option.

Accessing CATALOG MANAGER functions

The BMC Software Administrative products for DB2 panel (Figure 1) is provided to make it easy to start CATALOG MANAGER and interact with the other ISPF-based Administrative products.

**NOTE**

Product selections that are displayed on the panel depend on the products that you have installed.

---

Figure 1  Example of BMC Software Administrative Products for DB2 panel

<table>
<thead>
<tr>
<th>COMMAND ====&gt;</th>
<th>BMC Software Administrative Products for DB2------------------</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DASD MANAGER for DB2</td>
<td>Manage or monitor DB2 physical objects</td>
</tr>
<tr>
<td>2 CATALOG MANAGER for DB2</td>
<td>Execute DDL or query DB2 Catalog</td>
</tr>
<tr>
<td>3 CHANGE MANAGER for DB2</td>
<td>Manage changes to DB2 objects/structures</td>
</tr>
</tbody>
</table>

DB2 SSID . . . DECA (? = SSID List)
DB2 Catalog Access . . DIRECT (Direct.Indirect)
Use Shared or Individual product ISPF APPLID? S (S/I - Admin Products only)
To start CATALOG MANAGER

This procedure describes how to start CATALOG MANAGER from the BMC Software Administrative Products for DB2 panel (Figure 1).

1 On the Command line, type the number that corresponds to the CATALOG MANAGER for DB2 option.

2 In the DB2 SSID field, type the SSID or group attach name to which you want to attach.

--- NOTE ---

The SSID that is specified here cannot be the group SSID that is used for data sharing in a sysplex.

3 In the DB2 Catalog Access field, type the method to be used to attach to the DB2 subsystem:

- DIRECT indicates that you will attach directly to a real subsystem.
  
  If the DB2 SSID to which you are attaching uses an ASCII encoding scheme, you must use the DIRECT method.

- INDIRECT indicates that you will attach to an installed copy or view of the specified subsystem catalog.

4 In the Use Shared or Individual product ISPF APPLID field, specify one of the following types of ISPF Application IDs (APPLID):

- To specify an APPLID for each DB2 subsystem, type S (shared).
- To specify an APPLID for each product, type I (individual).

5 Press Enter.

The CATALOG MANAGER Primary Menu panel is displayed (Figure 2).
Using the Primary Menu panel

The Primary Menu panel is your starting point for accessing the many functions and features of CATALOG MANAGER.

Using the Primary Menu panel

You can use the Primary Menu panel to generate a list of DB2 objects and to access other CATALOG MANAGER functions. The panel contains the elements that are listed in Table 1.

Table 1   Description of Primary Menu panel (Part 1 of 2)

<table>
<thead>
<tr>
<th>Panel element</th>
<th>Description or use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title line</td>
<td>includes the following information:</td>
</tr>
<tr>
<td></td>
<td>- SSID to which you are attached</td>
</tr>
<tr>
<td></td>
<td>- product name and version number</td>
</tr>
<tr>
<td></td>
<td>- panel title</td>
</tr>
<tr>
<td>Command line</td>
<td>user enters CATALOG MANAGER commands, TSO commands, and ISPF commands</td>
</tr>
<tr>
<td>Action field</td>
<td>user enters the corresponding number or letter of an action from the list of actions</td>
</tr>
</tbody>
</table>
Using the Command line

The Command line supports several types of input information, including BMC commands, ISPF commands, and TSO commands.

Use one of the following methods to issue a command from the Command line of the Primary Menu panel:

- Enter the number that corresponds to an Action. Actions are described in “Selecting an Action” on page 38.

- If you are familiar with the appropriate command syntax, invoke an Action by entering its equivalent command. Use the entire command or enough of the command to distinguish it from other commands.

**NOTE**

If you type a command on the Command line of the Primary Menu panel, do not specify an action in the Action field.

The commands that you can use depend on which panel is displayed. To display the Commands List panel, which lists the available CATALOG MANAGER commands for any panel, enter COMMAND (CMD) on the Command line. Available commands, their syntax specifications, and parameters are described in Appendix E, “Commands,” and in the online Help.

Using a command recognition character

Some CATALOG MANAGER commands, such as PRINT, are also ISPF commands. When such a command is entered on the Command line, ISPF assumes that the command is an ISPF command (provided that the command is defined in the ISPF CMD table) and processes it as such.
To force ISPF to ignore a command so that it can be passed to CATALOG MANAGER for processing, you must type the > command recognition character before the command with no intervening spaces (for example, >PRINT).

Selecting an Action

Each Action on the Primary Menu panel invokes a CATALOG MANAGER command. To choose an Action, perform one of the following actions:

- On the Command line, type the corresponding number.
- In the Action field, type the corresponding number.
- In the Action field, type the corresponding letter (shown in parentheses) that labels the desired action.

**NOTE**

When you select the following actions, you must also identify an object type:

- List using customizable lists
- Search for catalog objects
- Create objects
- Grant privileges

For more information, see “Identifying an object type” on page 38.

Whether you choose an Action by number or by letter, CATALOG MANAGER provides the command syntax. The LIST option is the default action for the Primary Menu panel.

Identifying an object type

The object type that you include in many commands identifies the set of DB2 objects with which you want to work. Figure 2 on page 36 and Table 2 show the DB2 object types that are displayed on the Primary Menu panel, their object type codes.

To choose an object type, type its two-letter code. For example, to create a list of all databases, type DB in the Obj type field and press Enter. CATALOG MANAGER generates a list of all databases in the current catalog.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Object types displayed on the Primary Menu panel (Part 1 of 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object type code</td>
<td>Object type</td>
</tr>
<tr>
<td>DB</td>
<td>database</td>
</tr>
<tr>
<td>SG</td>
<td>storage group</td>
</tr>
</tbody>
</table>
Identifying an object type

Table 2  Object types displayed on the Primary Menu panel (Part 2 of 2)

<table>
<thead>
<tr>
<th>Object type code</th>
<th>Object type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB</td>
<td>table</td>
</tr>
<tr>
<td>VW</td>
<td>view</td>
</tr>
<tr>
<td>IX</td>
<td>index</td>
</tr>
<tr>
<td>TS</td>
<td>table space</td>
</tr>
<tr>
<td>SU</td>
<td>system privilege user</td>
</tr>
<tr>
<td>SY</td>
<td>synonym</td>
</tr>
<tr>
<td>PG</td>
<td>package</td>
</tr>
<tr>
<td>CI</td>
<td>collection ID</td>
</tr>
<tr>
<td>PL</td>
<td>plan</td>
</tr>
<tr>
<td>AL</td>
<td>alias</td>
</tr>
<tr>
<td>US</td>
<td>user</td>
</tr>
<tr>
<td>CO</td>
<td>column</td>
</tr>
<tr>
<td>DM</td>
<td>DBRM</td>
</tr>
<tr>
<td>ST</td>
<td>string</td>
</tr>
<tr>
<td>LO(^a)</td>
<td>location</td>
</tr>
<tr>
<td>CK</td>
<td>check constraint</td>
</tr>
<tr>
<td>PR</td>
<td>stored procedure</td>
</tr>
<tr>
<td>XT</td>
<td>auxiliary table(^b)</td>
</tr>
</tbody>
</table>

\(^a\) The product displays the object type on the panel if the Distributed Data Facility (DDF) is defined to CATALOG MANAGER.

\(^b\) An auxiliary table contains a single large object (LOB) column. An auxiliary table resides in an auxiliary (or LOB) table space.

Table 3 lists the DB2 object types that are supported but are not displayed on the Primary Menu panel. Unless otherwise noted, you can enter the code in the **Obj Type** field or on the **Command** line on the Primary Menu panel. Press HELP from the Primary Menu panel to list all supported object types.

Table 3  Object types not displayed on the Primary Menu panel (Part 1 of 3)

<table>
<thead>
<tr>
<th>Object type code</th>
<th>Object type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AU</td>
<td>authorization(^a)</td>
</tr>
<tr>
<td>BP</td>
<td>buffer pool</td>
</tr>
<tr>
<td>CA</td>
<td>column authorization</td>
</tr>
<tr>
<td>CD</td>
<td>check dependent</td>
</tr>
<tr>
<td>CL</td>
<td>column label(^a)</td>
</tr>
<tr>
<td>CP</td>
<td>constraint dependent</td>
</tr>
<tr>
<td>CX</td>
<td><em>(DB2 Version 9 or later)</em> trusted context</td>
</tr>
<tr>
<td>Object type code</td>
<td>Object type</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CXA</td>
<td>(DB2 Version 9 or later) trusted context authorization IDb</td>
</tr>
<tr>
<td>CXT</td>
<td>(DB2 Version 9 or later) trusted context attributejb</td>
</tr>
<tr>
<td>C2</td>
<td>check constraint (for tables created in DB2 Version 7 or later)</td>
</tr>
<tr>
<td>DP</td>
<td>(DB2 Version 9 or later) dependency</td>
</tr>
<tr>
<td>DS</td>
<td>data setda</td>
</tr>
<tr>
<td>DT</td>
<td>data or distinct type</td>
</tr>
<tr>
<td>EN</td>
<td>(DB2 Version 9 or later) environment variables</td>
</tr>
<tr>
<td>FK</td>
<td>foreign keya</td>
</tr>
<tr>
<td>FN</td>
<td>function routine</td>
</tr>
<tr>
<td>FO</td>
<td>routine optionec</td>
</tr>
<tr>
<td>FP</td>
<td>routine parameter</td>
</tr>
<tr>
<td>FS</td>
<td>routine source</td>
</tr>
<tr>
<td>IC</td>
<td>image copya</td>
</tr>
<tr>
<td>IL</td>
<td>IP list</td>
</tr>
<tr>
<td>IM</td>
<td>index mixedda</td>
</tr>
<tr>
<td>IN</td>
<td>IP nameed</td>
</tr>
<tr>
<td>IP</td>
<td>index space partition</td>
</tr>
<tr>
<td>IS</td>
<td>index space</td>
</tr>
<tr>
<td>ISS</td>
<td>index space statisticsefe</td>
</tr>
<tr>
<td>JB</td>
<td>jar object</td>
</tr>
<tr>
<td>JC</td>
<td>jar contents</td>
</tr>
<tr>
<td>JP</td>
<td>Java option</td>
</tr>
<tr>
<td>JT</td>
<td>(DB2 Version 9 or later) Java path</td>
</tr>
<tr>
<td>KC</td>
<td>key columna</td>
</tr>
<tr>
<td>KT</td>
<td>(DB2 Version 9 or later) key-target</td>
</tr>
<tr>
<td>KTD</td>
<td>(DB2 Version 9 or later) key-target distributionf</td>
</tr>
<tr>
<td>KTH</td>
<td>(DB2 Version 9 or later) key-target historyf</td>
</tr>
<tr>
<td>KTS</td>
<td>(DB2 Version 9 or later) key-target statisticsf</td>
</tr>
<tr>
<td>KU</td>
<td>key column user</td>
</tr>
<tr>
<td>LK</td>
<td>limit keyag</td>
</tr>
<tr>
<td>LL</td>
<td>LU listda</td>
</tr>
<tr>
<td>LM</td>
<td>LU modeed</td>
</tr>
<tr>
<td>LS</td>
<td>LU modeselectda</td>
</tr>
<tr>
<td>LU</td>
<td>LU namesed</td>
</tr>
<tr>
<td>MQT</td>
<td>materialized query tablec</td>
</tr>
<tr>
<td>MX</td>
<td>mixed object typesa</td>
</tr>
</tbody>
</table>
### Table 3  Object types not displayed on the Primary Menu panel (Part 3 of 3)

<table>
<thead>
<tr>
<th>Object type code</th>
<th>Object type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>native SQL procedure</td>
</tr>
<tr>
<td>OB</td>
<td>online schema changes</td>
</tr>
<tr>
<td>OS</td>
<td>LOBSTATS</td>
</tr>
<tr>
<td>PA</td>
<td>plan authorization</td>
</tr>
<tr>
<td>PI</td>
<td>packlist</td>
</tr>
<tr>
<td>PK</td>
<td>primary key</td>
</tr>
<tr>
<td>PT</td>
<td>index and table space partition</td>
</tr>
<tr>
<td>RD</td>
<td>(DB2 Version 9 or later) object role dependency</td>
</tr>
<tr>
<td>RE</td>
<td>relation</td>
</tr>
<tr>
<td>RI</td>
<td>referential integrity</td>
</tr>
<tr>
<td>RO</td>
<td>(DB2 Version 9 or later) roles</td>
</tr>
<tr>
<td>SC</td>
<td>schema</td>
</tr>
<tr>
<td>SE</td>
<td>identity column</td>
</tr>
<tr>
<td>TC</td>
<td>table constraint</td>
</tr>
<tr>
<td>TM</td>
<td>table space mixed</td>
</tr>
<tr>
<td>TP</td>
<td>table space partition</td>
</tr>
<tr>
<td>TR</td>
<td>trigger</td>
</tr>
<tr>
<td>TSS</td>
<td>table space statistics</td>
</tr>
<tr>
<td>TT</td>
<td>table space set</td>
</tr>
<tr>
<td>UA</td>
<td>user authorization</td>
</tr>
<tr>
<td>UN</td>
<td>user name</td>
</tr>
<tr>
<td>VL</td>
<td>volume</td>
</tr>
<tr>
<td>XR</td>
<td>(DB2 Version 9 or later) XML relationship</td>
</tr>
<tr>
<td>XS</td>
<td>(DB2 Version 9 or later) XML string</td>
</tr>
</tbody>
</table>

a  This object code is not valid in the **Obj type** field of the Primary Menu panel.

b  This object code is also available from a CX list.

c  This object code is valid for DB2 Version 8 or later.

d  This object code is valid if DDF is defined to CATALOG MANAGER.

e  This object code is also available from an IS list.

f  This object code is also available from a KT list.

g  This object code is valid for displaying limit keys on a table that uses table-controlled partitioning.

h  This object code displays only the tables in the table space that have referential integrity.
Specifying a qualifier

Table 4 lists the four-character object type codes that are available from the Command line of a KT list. These codes are valid for DB2 Version 9 or later.

<table>
<thead>
<tr>
<th>Object type code</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>KTDH</td>
<td>key-target distribution history</td>
</tr>
<tr>
<td>KTDS</td>
<td>key-target distribution statistics history</td>
</tr>
</tbody>
</table>

### Specifying a qualifier

You can further describe the object type that you want to list by specifying an object qualifier in the **Qualifier** field of the CATALOG MANAGER Primary Menu panel (Figure 2 on page 36).

An object qualifier is a character string that names a particular object or group of objects. You can specify all or any part of the name of the object in the **Qualifier** field. When you use an object qualifier, CATALOG MANAGER accesses only those objects that match the qualified name. Executing lists with qualifiers improves performance by reducing the time that it takes for CATALOG MANAGER to return the results that you want.

If you are familiar with the command syntax, you can include the object type and qualifier on the **Command** line following the command.

### Including wildcards in qualifiers

To generate a list of objects that match more than one character string, you can include a wildcard character in the qualifier. **Table 5** describes the wildcard characters that CATALOG MANAGER supports.
Specifying a qualifier

Chapter 2 Getting started with CATALOG MANAGER

Wildcards with fixed-length CHAR columns

In DB2, and therefore in CATALOG MANAGER, the position of a wildcard in the qualifier is important when the qualifier represents a fixed-length CHAR column. Table 6 describes the matches that result when such a qualifier contains wildcards in the following positions:

- before and after characters
- after characters
- before characters

In Table 6, the dots in the column headings represent any character, including blanks.

<table>
<thead>
<tr>
<th>Wildcard character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>% (percent sign)</td>
<td>matches any string of zero or more characters</td>
</tr>
<tr>
<td>* (asterisk)</td>
<td>For example, a qualifier value of AB%D or AB*D matches ABCD, AB123D, and ABD, but not AB.</td>
</tr>
<tr>
<td>? (question mark)</td>
<td>matches any single character</td>
</tr>
<tr>
<td>_ (underscore)</td>
<td>For example, a qualifier value of AB?D or AB_D matches ABCD, AB1D, but not ABD or AB12D.</td>
</tr>
</tbody>
</table>

**Note:** If the _ Wild switch is set to N, an underscore will not be considered as a wildcard character if no other wildcard characters (such as % or *) are included in the qualifier for a table list. For more information about switches, see “Setting CATALOG MANAGER switches” on page 95.

<table>
<thead>
<tr>
<th>Wildcard character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%ABC%</td>
<td>Yes</td>
</tr>
<tr>
<td>ABC%</td>
<td>No</td>
</tr>
<tr>
<td>%ABC</td>
<td>No</td>
</tr>
<tr>
<td>%ABC%</td>
<td>Yes</td>
</tr>
<tr>
<td>ABC%</td>
<td>Yes</td>
</tr>
<tr>
<td>%ABC</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Using nonprintable or nonviewable characters

**TIP**

In fixed-length CHAR columns, to find matches for values that have blanks, build the qualifier as follows:

- Delimit the qualifier with quotes.
- Use blanks in the qualifier in the same positions as the blanks in the value for which you are searching.

For example, to find a match in an eight-character fixed-length CHAR column for the pattern `xxxABCbb` (where b represents a blank), use the qualifier `%ABCbb`.

**Qualifiers for objects with two-part names**

CATALOG MANAGER uses the rules shown in Table 7 to match qualifiers for objects with two-part names, such as tables and indexes.

**Table 7  How CATALOG MANAGER qualifies two-part object names**

<table>
<thead>
<tr>
<th>If the qualifier string or wildcard pattern</th>
<th>CATALOG MANAGER attempts to match</th>
</tr>
</thead>
<tbody>
<tr>
<td>is in two parts separated by a period</td>
<td>the complete two-part object name</td>
</tr>
<tr>
<td>is followed by a period</td>
<td>the first part of the two-part object name</td>
</tr>
<tr>
<td>does not include a period</td>
<td>the second part of the two-part object name</td>
</tr>
</tbody>
</table>

**Qualifier exceptions**

Table 8 shows usage of qualifiers that might be unexpected.

**Table 8  Qualifier exceptions**

<table>
<thead>
<tr>
<th>Object type</th>
<th>Object type code</th>
<th>Note about qualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>table space partition</td>
<td>TP</td>
<td>Use the same qualifier as for a table space (TS).</td>
</tr>
<tr>
<td>index space partition</td>
<td>IP</td>
<td>Use the same qualifier as for an index (IX).</td>
</tr>
<tr>
<td>constraint dependency</td>
<td>CP</td>
<td>The full qualifier is <code>DTBCREATOR.DTBNAME</code>.</td>
</tr>
</tbody>
</table>

**Using nonprintable or nonviewable characters**

You must specify nonprintable and nonviewable characters (such as null and control characters) in limit keys, view text, trigger text, or check constraint text in an external hexadecimal format. These characters are not indicated in literal strings. For example, the EBCDIC string ‘2 1’ does not indicate that the second character is null; the string appears to be blank. The string should be specified as `XF200F1`.
Table 9 provides the hexadecimal formats for common nonprintable or nonviewable characters.

**Table 9  Nonprintable or nonviewable characters**

<table>
<thead>
<tr>
<th>Character</th>
<th>Hexadecimal format</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td>X'00'</td>
</tr>
<tr>
<td>horizontal tab ('HT')</td>
<td>X'05'</td>
</tr>
<tr>
<td>form feed</td>
<td>X'0C'</td>
</tr>
<tr>
<td>carriage return</td>
<td>X'0D'</td>
</tr>
<tr>
<td>new line ('NL')</td>
<td>X'15'</td>
</tr>
<tr>
<td>line feed ('LF')</td>
<td>X'25'</td>
</tr>
</tbody>
</table>

**Using ISPF commands**

Most of the ISPF commands in CATALOG MANAGER work the same as they do in other ISPF applications. Table 10 describes the most commonly used ISPF commands.

**Table 10  ISPF commands (Part 1 of 2)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANCEL</td>
<td>returns to the previous panel without saving any change that you made on the current panel</td>
</tr>
<tr>
<td>DOWN (or F8)</td>
<td>scrolls the panel down</td>
</tr>
<tr>
<td></td>
<td><strong>More:</strong> + on a panel indicates that more information is available below the current line.</td>
</tr>
<tr>
<td></td>
<td>Scrolling is available on a Model 2 3270 mainframe terminal, which uses a 24-line by 80-column display.</td>
</tr>
<tr>
<td>END (or F3)</td>
<td>validates and processes information, the same as the Enter key</td>
</tr>
<tr>
<td></td>
<td>In some panels, pressing END returns to the previous panel.</td>
</tr>
<tr>
<td>ENTER</td>
<td>processes information that is typed on the panel and executes any specified commands</td>
</tr>
<tr>
<td></td>
<td>For a sequence of related panels, pressing Enter validates the information on the current panel and displays the next panel in the sequence.</td>
</tr>
<tr>
<td>HELP (or F1)</td>
<td>provides panel-level Help</td>
</tr>
<tr>
<td>LEFT (or F10)</td>
<td>scrolls the panel to the left</td>
</tr>
<tr>
<td></td>
<td><strong>More:</strong> &lt; on a panel indicates that more information is available to the left.</td>
</tr>
<tr>
<td></td>
<td>On the CATALOG ROW panel, when you press F10, the previous object in the object list is displayed.</td>
</tr>
</tbody>
</table>
Getting help

CATALOG MANAGER provides you with panel-level Help. To obtain Help for a panel, press HELP or enter HELP on the Command line. Some panels also offer field-level Help. To access field-level Help, place the cursor on a field and press HELP.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFSHOW</td>
<td>displays the active function keys</td>
</tr>
<tr>
<td></td>
<td>Some panels use every available line to display input variables. To display all variables, enter PFSHOW OFF on the Command line.</td>
</tr>
<tr>
<td>RETURN (or =X)</td>
<td>returns to the Primary Menu panel</td>
</tr>
<tr>
<td>RIGHT (or F11)</td>
<td>scrolls the panel to the right</td>
</tr>
<tr>
<td></td>
<td><strong>More:</strong> &gt; on a panel indicates that more information is available to the right. On the CATALOG ROW panel, when you press F11, the next object in the object list is displayed.</td>
</tr>
<tr>
<td>SPLIT (or F2)</td>
<td>divides the panel and displays the ISPF Primary Option Menu in the new panel</td>
</tr>
<tr>
<td></td>
<td>If you start the product on both panels, ensure that each product is at the same version, release, and maintenance level.</td>
</tr>
<tr>
<td>SSE</td>
<td>starts the BMC Simple Space Estimation (SSE) feature to estimate space requirements for table space or index objects</td>
</tr>
<tr>
<td>SWAP (or F9)</td>
<td>switches from one split panel to another</td>
</tr>
<tr>
<td>UP (or F7)</td>
<td>scrolls the panel up</td>
</tr>
<tr>
<td></td>
<td><strong>More:</strong> - on a panel indicates that more information is available above the current line. Scrolling is available on a Model 2 3270 mainframe terminal, which uses a 24-line by 80-column display.</td>
</tr>
<tr>
<td>ZOOM (or F4)</td>
<td>displays the full value of an object with a long name in a dialog or enables you to enter a name that is longer than 18 characters</td>
</tr>
<tr>
<td></td>
<td>In an object list, you must type S in the <strong>Cmd</strong> column to display the full value of an object with a long name.</td>
</tr>
<tr>
<td></td>
<td>The data editing function does not support the F4 key.</td>
</tr>
</tbody>
</table>
Navigation in CATALOG MANAGER

In CATALOG MANAGER, you can move from one function to another function in several ways. All transactions usually begin at the Primary Menu panel. Figure 3 shows some of the more common paths through the CATALOG MANAGER panels.

Figure 3  Navigating through CATALOG MANAGER

Generating lists in CATALOG MANAGER

The LIST function can be considered the most fundamental feature of CATALOG MANAGER. It enables you to generate lists of objects from the DB2 catalog; you can then manipulate those objects easily by issuing CATALOG MANAGER and DB2 commands against the objects. Creating lists is basic to using CATALOG MANAGER efficiently.
Using a list panel

The following features are included on a list panel (Figure 4):

**Figure 4  Object list**

![Object list](image-url)

- the panel title line, which includes the following information:
  - the SSID to which you are attached
  
  You can change this field to indicate a server connection.
  
  - an indicator of whether catalog indirection is in effect
    
    - I indicates that you are using the indirect catalog
    - R indicates that you are using the real catalog
    - S indicates that you are connected to a server rather than the local DB2 subsystem
  
  - the panel title
  
  - the first row number displayed and the number of rows that completes the entire list

- the **Command** line, on which you enter CATALOG MANAGER commands, TSO commands, or ISPF commands
Generating an object list from the Primary Menu panel

- a stack level indicator below the scroll increment indicator

In Figure 4, the stack level indicator 01 means that this list was generated from the Primary Menu panel. Such a list is sometimes referred to as a level-one list.

When you generate subsequent lists from lists, CATALOG MANAGER increments the stack level indicator to show how many lists away from the Primary Menu you have navigated.

A list that is generated from another list is called a secondary list. You can generate a secondary list from a level-one list or from another secondary list. The instruction area of a secondary list panel indicates the source object of the previous list as the qualifier that was used to generate the secondary list. For more information about secondary lists, see “Generating secondary lists” on page 50.

- an instruction area that shows available commands, available lists, and other helpful information

- a Cmd column from which you can initiate a task for one or more objects in the list

You can type CATALOG MANAGER commands or object type codes in the Cmd column to generate secondary lists or access more information about the objects on the current list.

Generating an object list from the Primary Menu panel

If you cannot generate a list for a specific database or table, verify your DB2 authorization status with your system administrator, or generate a user authorization list for the catalog table and review your user authorizations.

To generate an object list from the Primary Menu panel

1 (optional) On the Command line or in the Action field, type L.

This step is optional because List using customizable lists is the default option. You can choose to leave the Command line and Action field blank when you generate a list from the Primary Menu.

2 In the Obj type field, type DB to specify a list of databases.

3 In the Qualifier field, type an identifying character string for the databases that you want to list.
Generating secondary lists

4 Press Enter.

CATALOG MANAGER lists the qualified databases:

- Pressing F11 scrolls to the right to display additional columns; pressing F10 scrolls to the left.
- The columns in the list correspond to the columns in the catalog table.
- The Max Lines per list value on the Options panel determines the number of rows that CATALOG MANAGER displays in a list. For more information, see “Setting basic options” on page 79.
- To view the column values for a single row, enter S in the Cmd field.

Generating secondary lists

This procedure describes how to generate secondary lists of objects.

To generate a table space list from a database list

After you have generated a list, you can use the objects on the list as sources for generating other lists.
1 In the **Cmd** column next to the source object (in this case, the database name) on the level-one list, type **TS** (Figure 5).

**Figure 5  Generating a table space list from a database list**

```
DEFF-R ------------------------- DATABASE LIST -------------------------
Command ===>                                                  Scroll ===> CSR
01
CMD will show commands for this list. Type command and press ENTER
Lists: CA CK CL CO DB IC IM IS IX MQT MX OS PA PDD PG PL RE RI SG TB TR TS
LIKE QZU%
Cmd  Database Owner  Stogroup Buf Pool DBID  ROShr Type Group    Encode
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
QZUDAC  ASUQA    QZUALL  BPO       1622                        A
QZUDA1  ASUQA    QZUALL  BPO        402                        A
QZUDB1  ASUQA    SYSDEFLT BP0       1346                        E
QZUDB2  ASUQA    SYSDEFLT BP0       1343                        E
QZUDB3  ASUQA    SYSDEFLT BP0       1347                        E
QZUDB4  ASUQA    SYSDEFLT BP0       1348                        E
QZUDB5  ASUQA    SYSDEFLT BP0       1350                        E
QZUDB6  ASUQA    SYSDEFLT BP0       1351                        E
QZUDCF  ASUQA    SYSDEFLT BP0       615                        E
QZUCI  ASUQA    SYSDEFLT BP8K0     1352                        E
QZUCI15 RDHZE3  SYSDEFLT BP0       1377                        E
QZUCI19 ASUQA    SYSDEFLT BP0       1378                        E
QZUCI02 ASUQA    SYSDEFLT BP0       1356                        E
QZUCI03 ASUQA    SYSDEFLT BP0       1358                        E
QZUCI04 ASUQA    SYSDEFLT BP0       1359                        E
```

2 Press Enter.

CATALOG MANAGER displays a secondary list of the table spaces in the source database (Figure 6).

**Figure 6  Table Space List panel**

```
DEFF-R ------------------------- TABLESPACE LIST -------------------------  ROW 1 OF 5
Command ===>                                                  Scroll ===> CSR
02
CMD will show commands for this list. Type command and press ENTER
Lists: AL CA CL CO DB DS FK IC IM IS IX LK MQT MX NP OS PA PDD PG PL PR PT RE
QUALIFIER: DATABASE=QZUDA1
Cmd Tablespace      Owner  Segsz Bpool Prts  Tbls     ActivPg    Status Enc Ty
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
QZUDA1.QZUS01A1   ASUQA      4 BP0       0    1       130        A    A
QZUDA1.QZUS02A1   ASUQA     16 BP0       0    2       146        A    A
QZUDA1.QZUS03A1   ASUQA     16 BP0       0    1       1440       A    A
QZUDA1.QZUS04A1   ASUQA     64 BP0       0    2       540        A    A
QZUDA1.QZUS05A1   ASUQA     64 BP0       0    1       720        A    A
```
You can continue to generate secondary lists by typing an appropriate object type code in the **Cmd** column next to the source object on the current list. Valid object types are shown in the **LIST** section of the panel.

**To generate lists of tables within multiple table spaces**

You can use multiple objects as source objects.

1. In the **Cmd** column next to the source table space names, type TB (Figure 7).

**Figure 7 Generating table lists from multiple table spaces**

```
DEFF-R ------------------------  TABLESPACE LIST  ----------------- ROW 1 OF 5
Command ===>                                                  Scroll ===> CSR
CMD will show commands for this list.  Type command and press ENTER
Lists: AL CA CL CO DB DS FK IC IM IS IX LK MQT MX NP OS PA PDD PG PL PR PT RE
QUALIFIER: DATABASE=QZUDA1
Cmd Tablespace      Owner  Segsz Bpool Prts  Tbls     ActivPg    Status Enc Ty
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
TBQZUDA1.QZUS01A1   ASUQA      4 BP0       0    1       130        A    A
TBQZUDA1.QZUS02A1   ASUQA     16 BP0       0    2       146        A    A
QZUDA1.QZUS03A1   ASUQA      0 BP0       4    1      1440        A    A
QZUDA1.QZUS04A1   ASUQA     64 BP0       0    2       540        A    A
QZUDA1.QZUS05A1   ASUQA      0 BP0       4    1       720        A    A
******************************  BOTTOM OF DATA  ******************************
```

2. Press Enter.

CATALOG MANAGER first displays a secondary list of tables within the first source table space (Figure 8).
Generating secondary lists

Chapter 2 Getting started with CATALOG MANAGER

Figure 8   Table List panel for first source table space

3 Press END.

CATALOG MANAGER displays the list of tables in the next source table space (Figure 9).

Figure 9   Table List panel for next source table space

4 (optional) Press END to display each secondary list in succession.
To generate a list by specifying different object types

You can also generate secondary lists by specifying different object types for different source objects, as shown in the following example.

1. To generate a list of indexes in the first table (Figure 9) and a list of primary keys in the second table, complete the following steps.

   A. In the Cmd (C) column next to the first table name, type IX (index).

   B. In the Cmd (C) column next to the second table name, type KC (key columns) (Figure 10).

   C. Press Enter.

Figure 10 Generating lists of varied dependent objects

<table>
<thead>
<tr>
<th>CMD</th>
<th>Table Name</th>
<th>Database Tblspace</th>
<th>ColsPK</th>
<th>Type</th>
<th>Rows</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>IX</td>
<td>QZU.QZUT01_DA1S02</td>
<td>QZUDA1</td>
<td>QZUS02A1</td>
<td>T</td>
<td>2036</td>
<td>128</td>
</tr>
<tr>
<td>KC</td>
<td>QZU.QZUT02_DA1S02</td>
<td>QZUDA1</td>
<td>QZUS02A1</td>
<td>O</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

CATALOG MANAGER first displays a secondary list of the indexes in the first source table (Figure 11).
Generating secondary lists

Chapter 2 Getting started with CATALOG MANAGER

Figure 11 Index List panel for first source table

| DEFF-R -------------------------- | INDEX LIST %----------------- | ROW 1 OF 5 |
| Command ===> | Scroll ===> CSR |

CMD will show commands for this list. Type command and press ENTER
Lists: CO DB DS IC IM IP IS ISS IX KC KT KD KTD KTDS KTH KTS LK NP OB PDD PG
QUALIFIER: TABLE=QZU.QZUT01_DA1S02

<table>
<thead>
<tr>
<th>C</th>
<th>Index Name</th>
<th>Table Name</th>
<th>UT</th>
<th>Cl</th>
<th>Col Keys</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QZU.QZUX01_DA1S02T01</td>
<td>QZU.QZUT01_DA1S02</td>
<td>P2</td>
<td>NY</td>
<td>1 2036</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUX02_DA1S02T01</td>
<td>QZU.QZUT01_DA1S02</td>
<td>D2</td>
<td>NN</td>
<td>6 2036</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUX03_DA1S02T01</td>
<td>QZU.QZUT01_DA1S02</td>
<td>D2</td>
<td>NN</td>
<td>19 2036</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUX04_DA1S02T01</td>
<td>QZU.QZUT01_DA1S02</td>
<td>D2</td>
<td>YN</td>
<td>19 2036</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUX05_DA1S02</td>
<td>QZU.QZUT01_DA1S02</td>
<td>U2</td>
<td>NN</td>
<td>19 2036</td>
</tr>
</tbody>
</table>

****************************** BOTTOM OF DATA ******************************

Figure 12 Column List panel for second source table

| DEFF-R ------------------------ | KEY COLUMN LIST %---------------- | ROW 1 OF 2 |
| Command ===> | Scroll ===> CSR |

CMD will show commands for this list. Type command and press ENTER
Lists: CL CO KC
QUALIFIER: TABLE=QZU.QZUT02_DA1S02

<table>
<thead>
<tr>
<th>C</th>
<th>CmIndex Name</th>
<th>Column</th>
<th>Seq Num</th>
<th>O</th>
<th>Lth</th>
<th>DatTyp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QZU.QZUX01_DA1S02T02</td>
<td>EMPNO</td>
<td>1</td>
<td>2</td>
<td>A</td>
<td>9 CHAR</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUX01_DA1S02T02</td>
<td>SSSNO</td>
<td>2</td>
<td>1</td>
<td>A</td>
<td>9 CHAR</td>
</tr>
</tbody>
</table>

****************************** BOTTOM OF DATA ******************************

2 Press END.

CATALOG MANAGER displays a secondary list of the key columns in the second source table (Figure 12).

You can continue to generate lists, press END to display a previous list, or issue CATALOG MANAGER commands for the objects on the current list.
Listing and executing commands

Using CATALOG MANAGER, you can execute BMC utility commands, DB2 commands, DB2 utility commands, and JCL generation commands. Executing commands in CATALOG MANAGER requires minimum input. In most cases, you enter one command keyword; you might follow the keyword with one or more parameters. Most commands can be invoked by entering an abbreviated form of the command keyword that distinguishes it from similar keywords.

CATALOG MANAGER customizes command execution based on your installation defaults and option settings. Where appropriate, you can also modify the options that are used to complete an action.

For more information about commands, see Appendix E, “Commands.”

Displaying the online command reference

You can display a list that includes all commands and objects for which they are valid.

Displaying the complete online command reference

1 On the Command line of the Primary Menu panel, type COMMAND (CMD).

2 Press Enter.

   CATALOG MANAGER displays the Commands List panel for all commands (Figure 13).
Accessing information about a specific command

You can display information about a specific command from the Commands List panel, from the Primary Menu panel, or from an object list panel.

- To display the syntax and description for a command on the Commands List panel, complete the following steps:

  1. In the **Command** column to the left of the command, type `S`.
  2. Press **Enter**.

- To display the syntax and description for a command from the Primary Menu panel or an object list panel, complete the following steps:

  1. On the **Command** line, type `CMD` followed by a space and the command (for example, `CMD AL`).
  2. Press **Enter**.
Issuing command prefixes

Listing commands that are valid for an object list

To display a list of commands that are valid for a particular object list, complete the following steps:

1. On the Command line of the object list panel, type COMMAND (CMD).
2. Press Enter.

CATALOG MANAGER displays the Commands List panel for all commands that are valid for the source object.

Issuing command prefixes

In most situations, CATALOG MANAGER requires that you type only a command prefix, that is, as much of the command as is necessary to differentiate it from other commands. For example, you can type the command prefix LI to issue the LIST command because there are currently no other commands that begin with the letters LI. To issue the COPY command, however, you must type COPY in full to distinguish it from the COPYAUTHS command, which you can issue as COPYA.

NOTE
The DESCRIBE command is an exception to this requirement. The DES command produces a subset of the output that is provided by the DESC command.

Abbreviating command keywords and object names

You can abbreviate some commands and add an abbreviated object name. For example, you can combine the DISPLAY command with its valid object names: DISD for DISPLAY DATABASE, DIST for DISPLAY THREAD, and DISU for DISPLAY UTILITY.

Omitting object types from commands

If a command has one or more parameters, you can omit these parameters under certain circumstances. In such cases, CATALOG MANAGER supplies the missing values based on where you issued the command. For example, if you issue the SEARCH command from the Command line of a table space list without specifying an
object type, CATALOG MANAGER assumes that you want to display the search panel for table spaces. When you issue a command from the Command line of an object list panel, you can omit the object type (the default type is the same as the list) and enter an object name.

Including multiple objects in a command

Most commands have no limit to the number of objects that you can include. GRANT and REVOKE have no limit on table lists and plan lists, but on all other types of lists, these commands have a maximum of 21 items. The number of items allowed for a utility depends on the number of control statements executed by that utility. For example, BMC utilities allow one complete control statement for each job step, but other utilities can combine a set of control statements within a single job step. In either case, CATALOG MANAGER allows a maximum of 999 control statement sets to be executed.

NOTE

When you return to a list panel after executing a command, remnants of the command might still appear in the Command column. You can type over these characters to execute another command. However, you must be sure to erase any leftover characters and spaces remaining in the field.

To erase leftover characters, type the next command and press EOF to erase to the end of the field. Then press Enter.

Issuing commands from the Cmd column

You can also issue some commands from the Cmd (C) column. For example, to create a new table based on a source table, type CREATE (CR) in the Cmd column beside a table name and press Enter.

CATALOG MANAGER provides other commands for showing and printing information, such as catalog statistics, execution history for BMC utilities, object statistics, and space-estimation information.

Issuing Wait-for-Enter commands against multiple objects

Some CATALOG MANAGER commands are referred to as Wait-for-Enter (WFE) commands. These commands enable you to select multiple source objects in a list and process them collectively to save time.
You can issue commands against all objects in a list, or against sequential or nonsequential objects by using the following methods.

**Issuing a command against individually selected objects**

1. In the **Cmd (C)** column beside each source object name, type the command (Figure 14).

   **NOTE**
   To avoid possible conflicts, include a space after commands that you type in the **Cmd (C)** column when those commands overlap the value in the adjoining column.

   **Figure 14** Tablespace List panel with line commands for copying multiple objects

   ![Figure 14](image)

2. Press Enter.

**Issuing a shortcut command**

1. In the **Cmd (C)** column next to the first source object name, type the command.

2. Type an equal sign (=) to the left of the names of the other source objects (Figure 15).

   These objects must be of the same type as the object on which you issued the command.
3 Press Enter.

**Selecting all objects in the list**

To issue a command against all objects in the list, type the command followed by the keyword ALL on the Command line of the list panel (Figure 16).
Excluding listed objects before issuing commands

1 On the object list panel, in the Cmd (C) column next to each of the objects, mark the objects that you want to exclude by typing the designator X (Figure 17).

Figure 16  Tablespace List panel with command to copy all listed table spaces

```plaintext
DEFF-R ------------------------  TABLESPACE LIST  ----------------- ROW 1 OF 5
Command ===> BMCCOPY ALL   Scroll ===> CSR

02
CMD will show commands for this list. Type command and press ENTER
Lists: AL CA CL CO DB DS FK IC IM IS IX LK MQT MX NP OS PA PDD PG PL PR PT RE
QUALIFIER: DATABASE=QZUDA1

Cmd Tablespace      Owner  Segsz Bpool Prts  Tbls     ActivPg    Status Enc Ty
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
  QZUDA1.QZUS01A1   ASUQA      4 BP0       0    1       130        A    A
  QZUDA1.QZUS02A1   ASUQA     16 BP0       0    2       146        A    A
  QZUDA1.QZUS03A1   ASUQA     64 BP0       0    2       540        A    A
  QZUDA1.QZUS05A1   ASUQA     64 BP0       0    2       720        A    A

******************************  BOTTOM OF DATA  ******************************
```

Figure 17  Tablespace List panel with exclude commands

```plaintext
DEFF-R ------------------------  TABLESPACE LIST  ----------------- ROW 1 OF 5
Command ===>                                                  Scroll ===> CSR

02
CMD will show commands for this list. Type command and press ENTER
Lists: AL CA CL CO DB DS FK IC IM IS IX LK MQT MX NP OS PA PDD PG PL PR PT RE
QUALIFIER: DATABASE=QZUDA1

Cmd Tablespace      Owner  Segsz Bpool Prts  Tbls     ActivPg    Status Enc Ty
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
  QZUDA1.QZUS01A1   ASUQA      4 BP0       0    1       130        A    A
X QZUDA1.QZUS02A1   ASUQA     16 BP0       0    2       146        A    A
  QZUDA1.QZUS03A1   ASUQA     64 BP0       0    2       540        A    A
  QZUDA1.QZUS05A1   ASUQA     64 BP0       0    2       720        A    A

******************************  BOTTOM OF DATA  ******************************
```
Using Fast Path Navigation

The Installation System for the Administrative products provides a feature called Fast Path Navigation. This feature enables you to switch from one product to another and then return to the original product. To initiate Fast Path Navigation, enter the name of the product to which you want to switch on the Command line of the current product. For a list of the products and commands, see Table 11.

For example, if you are currently using CATALOG MANAGER and want to alter an object using CHANGE MANAGER, enter `BMCCHG` on the CATALOG MANAGER Command line of the Primary Menu panel or an object list panel. The main menu for the requested product is displayed. In this case, the CATALOG MANAGER session is temporarily suspended and then resumed when you exit CHANGE MANAGER.

For more information about enabling Fast Path Navigation, see the installation guide.

### Table 11 Fast Path Navigation Commands

<table>
<thead>
<tr>
<th>Product</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER</td>
<td>BMCALTER</td>
</tr>
<tr>
<td>CHANGE MANAGER</td>
<td>BMCCCHG</td>
</tr>
<tr>
<td>CATALOG MANAGER</td>
<td>BMCCAT</td>
</tr>
<tr>
<td>BMC DASD MANAGER PLUS for DB2</td>
<td>BMCDASD</td>
</tr>
</tbody>
</table>

**NOTE**

To enable Fast Path Navigation when locking options for data editing have been specified (see “Setting options for editing data” on page 231), the installer must enable the locking options command in the BMCADMF2 CLIST.

For more information about using lists, see Chapter 5, “Working with lists and searches.”
Where to go from here

In this chapter, you learned the basics of how to use CATALOG MANAGER to manage your DB2 catalog. Chapter 3, “Setting up CATALOG MANAGER,” shows you how to assess your CATALOG MANAGER installation and modify options for the appearance and operation of the product.
Setting up CATALOG MANAGER

This chapter contains the following topics:

Before you begin .......................................................... 66
Viewing settings ......................................................... 66
  Viewing information about the environment ..................... 67
  Viewing DB2 initialization parameters .............................. 67
  Viewing the CATALOG MANAGER common area .................. 67
  Viewing and updating DB2 special registers ....................... 67
Granting access to CATALOG MANAGER functions ................. 69
  Using Execution plans ............................................. 70
  Manipulating plans .................................................. 71
Setting the MEMLIMIT system parameter ......................... 71
Using options to control your environment ......................... 71
  Using the installation options .................................... 72
  Using the user options .............................................. 72
  Using the product options ......................................... 73
  Putting it all together .............................................. 75
Setting user options .................................................. 76
  Defining an options data set ...................................... 77
  Setting basic options .............................................. 79
  Setting general options ........................................... 82
  Setting object use options ........................................ 85
  Setting JCL Generation options ................................... 86
  Setting data set options .......................................... 88
  Setting SQL and confirm options ................................ 90
  Setting SQL SELECT options ...................................... 93
  Setting panel graphic options .................................... 94
  Setting CATALOG MANAGER switches .......................... 95
  Setting DESCRIBE options ........................................ 98
Setting product options ............................................ 99
  Setting the JCL options for job cards ......................... 101
  Setting the JCL options for STEPLIBs .......................... 104
  Setting the JCL options for static data sets ..................... 105
  Setting the JCL options for tapes ................................ 109
  Setting the JCL options for temporary work data sets .......... 112
  Setting the JCL options for permanent data sets ............... 114
Before you begin

This chapter explains how to set options for the appearance and operation of CATALOG MANAGER. Some of these tasks are available to installers or administrators only. Check with your system administrator or database administrator to verify which user-defined settings to change.

Viewing settings

CATALOG MANAGER provides commands that make it easy for you to view or change how it works. You can also use CATALOG MANAGER commands to view DB2 special registers and initialization parameters.
**Viewing information about the environment**

From the Primary Menu panel or any list panel, you can enter the ENVIRONMENT (ENVI) command on the **Command** line to display information about the version of CATALOG MANAGER that you are running. This command provides information such as the CATALOG MANAGER version number, installation options module name, and command module name. The command also lists the PTFs that have been applied to the product and the product components.

**Viewing DB2 initialization parameters**

On the Primary Menu panel or any list panel, you can enter the DSNZPARM command on the **Command** line to display the DB2 DSNZPARM and DSNHDECP initialization parameter values.

The DSNZPARM parameters are used in the DB2 DSN6 macros. These macros were assembled to form the DSNZPARM used at initialization of the DB2 system to which CATALOG MANAGER is currently attached.

In the DSNZPARM display, CATALOG MANAGER shows the values set in DSNHDECP. Section DSNHDECP is read from the DSNEXIT library and must exist in the STEPLIB or be linklisted to process correctly.

**Viewing the CATALOG MANAGER common area**

From the Primary Menu panel or any list panel, you can enter PEEK on the **Command** line to display the CATALOG MANAGER common area in dump format. The common area stores many of the internal values for the CATALOG MANAGER installation. This display might be helpful if you must contact BMC Customer Support for some classes of problems.

**Viewing and updating DB2 special registers**

CATALOG MANAGER enables you to view and modify DB2 special registers. To display the DB2 Special Registers panel (Figure 18), from the Primary Menu panel or any object list panel, enter the SEE command on the **Command** line.
You can update the values of the following fields on the DB2 Special Registers panel by typing over the existing value:

- Current SQLID
- Current Degree
- Local LC_CTYPE
- Table Types for Optimization
- Current Optimization Hint
- Current Path
- Current Precision
- Refresh Age
- Current Rules

NOTE

When the value for the Current Package Set field is blank, all packages in the packlist are available to the user.
Granting access to CATALOG MANAGER functions

You can grant and restrict access to certain functions of CATALOG MANAGER through plan authorizations. In general, the plans grant only the authority to access and view catalog data. Table 12 lists the plans that CATALOG MANAGER provides. In addition to the product code for CATALOG MANAGER (ACT), the plan names include the version number (v) and the release number (r).

### Table 12  CATALOG MANAGER plans (Part 1 of 2)

<table>
<thead>
<tr>
<th>Plan name</th>
<th>Function name</th>
<th>Plan description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTvrDB</td>
<td>Bind and Rebind</td>
<td>allows access to binding and rebinding functions</td>
</tr>
</tbody>
</table>
| ACTvrDE   | Data Editing and Browsing | enables access to data editing and browsing functions  
This plan does not override DB2 table authorizations. |
| ACTvrDG   | Generate SQL | allows SQL statement generation |
| ACTvrDH   | Utility Status Display | enables displaying the status of BMC utilities  
ACTvrDH allows deleting rows from the BMC utility history table. Grant EXECUTE authority on this plan to users who should be able to display or terminate BMC utilities. |
| ACTvrDK   | Command Generation and Execution | enables generating and executing DB2 operator commands  
Grant EXECUTE authority on this plan to users who should be able to issue DB2 operator commands, such as START, STOP, DISPLAY, and TERM. |
| ACTvrDL   | Log Table Maintenance | allows browsing rows in the CATALOG MANAGER Audit, Session, and Drop Recovery Logs  
ACTvrDL allows deleting rows from CATALOG MANAGER logs. Grant EXECUTE authority on this plan to users who are responsible for administering CATALOG MANAGER. Grant SELECT authority to users who need to browse the logs, and grant DELETE authority to users who need access to purge functions. |
| ACTvrDM   | Display DB2 Catalog and SQL Information | displays information about the DB2 catalog and SQL generation and execution  
This plan allows the minimum access that is required to use CATALOG MANAGER. |
CATALOG MANAGER does not bypass any DB2 security when it generates and executes SQL, DML, or DB2 commands. DB2 rejects any action requested by CATALOG MANAGER for which the user is not authorized by DB2.

**NOTE**

DB2 requires that users have at least SELECT authorization to access catalog tables. The CATALOG MANAGER installation options settings cannot override the DB2 SELECT authorization requirement.

### Using Execution plans

In CATALOG MANAGER, you can execute a worklist through the Execution component by using the plans provided with the BMC ALTER, CHANGE MANAGER, or DASD MANAGER PLUS products when the following requirements are met.

- **ALTER, CHANGE MANAGER, or DASD MANAGER PLUS is installed.**

- The CATALOG MANAGER AOPTS installation option or BOPTS installation option specifies the installation options module name for ALTER, CHANGE MANAGER, or DASD MANAGER PLUS, as follows:

  — To use the ALTER execution plans, in CATALOG MANAGER specify the name of the ALTER installation options module for the AOPTS installation option.

  — To use the CHANGE MANAGER execution plans, in CATALOG MANAGER specify the name of the CHANGE MANAGER installation options module for the AOPTS installation option.

  — To use the DASD MANAGER PLUS execution plans, in CATALOG MANAGER specify the name of the DASD MANAGER PLUS installation options module for the BOPTS installation option.

### Table 12  CATALOG MANAGER plans (Part 2 of 2)

<table>
<thead>
<tr>
<th>Plan name</th>
<th>Function name</th>
<th>Plan description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTvrDS</td>
<td>Search</td>
<td>enables search functions</td>
</tr>
<tr>
<td>ACTvrDU</td>
<td>Grant Authorities and</td>
<td>enables generating and submitting JCL for BMC and IBM utilities</td>
</tr>
<tr>
<td></td>
<td>Submit BMC Utilities or</td>
<td>Grant EXECUTE authority on this plan to users who should be able to</td>
</tr>
<tr>
<td></td>
<td>IBM Utilities</td>
<td>to grant authorities or submit BMC or IBM utilities.</td>
</tr>
</tbody>
</table>

---

---

---

### Table 12  CATALOG MANAGER plans (Part 2 of 2)
For more information about executing worklists, see the following documentation:

- *ALTER and CHANGE MANAGER for DB2 User Guide*
- *DASD MANAGER PLUS for DB2 User Guide*

### Manipulating plans

CATALOG MANAGER provides commands such as FREE, BIND, REBIND, and PACKIT for manipulating plans. For more information, press HELP on a panel where these commands are valid, or see Appendix E, “Commands.”

### Setting the MEMLIMIT system parameter

CATALOG MANAGER requires above-the-bar memory and might abend if sufficient memory is not available. The default value for the System Management Facility (SMF) MEMLIMIT parameter is 2 GB. This value is set in member SMFPRMxx in SYS1.PARMLIB. Use any of the following methods if you need to override the default value:

- Specify the MEMLIMIT parameter in the JCL.
- Specify REGION=0M in the JCL.
- Use the SMF IEFUSI exit.

### Using options to control your environment

For your production requirements, you should reevaluate the option settings for CATALOG MANAGER. For most situations, the installation default values are sufficient. However, you might want to customize the operating environment and panels.

CATALOG MANAGER uses default, user, and product options to define the operating environment and to specify how the product’s components work. The options also contain default values for data set names and allocations, job control language (JCL) generation information, and component plan names.

These options provide you with the ability to

- tailor the interface
- set up defaults for generating job statements
- specify defaults for parameters, names, and prefixes for allocated data sets
Typically, the person who installs CATALOG MANAGER sets default values for user options. For information about establishing installation option values at installation, see the installation guide.

**Using the installation options**

The default operating environment is controlled by a number of option values that are defined in the installation options module. The components of CATALOG MANAGER use the global values that are stored in the installation options module to determine how to process information. The Installation System generates the installation options module when you install CATALOG MANAGER. The module contains an assembly-language program with an options macro.

You can customize CATALOG MANAGER for all users by editing the default values in the installation options module. The default name of the CATALOG MANAGER module is ACTDOPD1. The source of the installation option modules is located in the HLQ.UDBCNTL data set. HLQ identifies the high-level qualifier that you specify when you install the products.

**NOTE**

CATALOG MANAGER uses plan names directly. If you need to specify different plan name values for each DB2 subsystem, you must have multiple installation option modules.

From the Primary Menu panel or any list panel, enter `DOPTS` on the Command line to display the installation options that are in effect for the current CATALOG MANAGER session. For more information about the installation options, see the installation guide.

**Using the user options**

The first time that you run CATALOG MANAGER, the product creates an options data set and copies the values from your ISPF variables or the installation options module into the data set. The values are stored in this data set in XML format and are referred to as your *user options*. The product uses these user options to generate JCL and to generate keywords for an input stream for each user’s subsystem. The product uses the AEXIN input stream, which is used by the Execution component.
You can refresh the values in your user options by editing and reassembling the installation options module. To refresh an option value in all existing ISPF profiles, type a comma and an \texttt{R} after the option value and then enclose the value in parentheses, as shown in the following example:

\begin{verbatim}
DBCS=(N,R),
\end{verbatim}

\textbf{NOTE}

Do not remove the comma after the right parenthesis or the continuation character (*) in column 72, except for the last option value.

The next time that you run the product, the new global value replaces the old local value in the user options. You can modify the local value through the options panels. If you need to change the installation options after installation, you must reassemble the installation options module.

For more information about setting your user options, see “Setting user options” on page 76. For more information about refreshing user options, see the installation guide.

\section*{Using the product options}

The POFDS keyword (in the installation options module) specifies an 80-character sequential file. This product options file (POF) is built during product installation and contains parameters and values for the JCL Generation options. The file is located in the \texttt{HLQ.UDBCNTL} data set. The POF does not require assembly and linkage and does not need to reside in an APF-authorized data set.

When you install the products, only one POF is created. This initial POF is initialized and populated with the default ISPF variables and values from the installation panels. This POF is shared among several products, if those products are installed at the same time.

In addition, the Installation System will use the same application ID (or profile) for the products in the BMCDB2 CLIST. This single application ID enables the JCL Generation options to be shared with other products, such as the BMC CHANGE MANAGER product. Thus, when you specify an option for generating JCL in one product, your selection applies to all of the products. Although BMC recommends that you use a single application ID, you can choose individual product application IDs on the BMCDB2PR panel.
JCL Generation also handles user POFs, which are POFs that can be written from the
ISPF variables that are set in CATALOG MANAGER or edited. You can use a user
POF to reset all of the options that you will use in the current session to create JCL.
You can also use the user POFs to set options for different sets of applications,
particularly if the applications have different naming standards.

**NOTE**

If a POF keyword in your user POF uses a library from an earlier version of the product,
update the keyword to use a library for the most recently installed version of the product. For example, assume that the value of the BMC_COPY_LOAD keyword is
BMC9300.ACM.D91.LOAD and then you installed version 10.1.00 of the product. Update the
value of the keyword to a version 10.1.00 LINK library (BMC1010.ACM.D10.DBLINK).

JCL Generation uses the variables in the ISPF profile when generating JCL. When you
start CATALOG MANAGER, JCL Generation determines whether to reset the
variables in the ISPF profile:

- The first time that the product is invoked, all of the values in the ISPF profile are
  set to the values that are in the initial POF. If a POF is not specified, default values
  are assigned to the variables in the profile.

- If the POFDATE parameter in the initial POF is greater than the value of the POF
date that is stored in the ISPF profile, the values in the POF that are marked with
  refresh ,(R) are used to reset the ISPF variables.

- If you specify a new initial POF in the POFDS installation option, the values in the
  POF that are marked with refresh ,(R) are used to reset the ISPF variables. The
  value of the POFDATE keyword in the new initial POF is saved in the ISPF profile.

For more information about setting the POF options, see “Setting product options” on
page 99. For more information about POFs, see the installation guide.
Putting it all together

Figure 19 illustrates how CATALOG MANAGER uses the installation options, user options, product options, and override options.

**Figure 19  How CATALOG MANAGER uses options**
Setting user options

CATALOG MANAGER provides a variety of options that enable you to control certain operations such as specifying general preferences for data set names, setting the parameter values that control JCL, DDL, and SQL processing, and customizing the panel highlight and color options.

In general, panels for each group of options are displayed in sequence. On most of the panels, follow this general procedure to set values for the options that are shown.

1. Type over the current value. The last value specified is the one that is saved unless you discard the changes.

2. After you view or change the values, you can continue by using one of the following methods:

   - Press END to save the changes and return to the Options panel. The values are stored in your profile for use in the current and future sessions until you change them again.

   - To exit without saving any changes, use the CANCEL command on the Command line of the Options panel, and then press Enter.

To display a help panel that describes the fields appearing on any options panel, press HELP, or enter HELP on the Command line of the appropriate options panel. In addition to field descriptions, the help panels show the commands that you can use on the panel.

CATALOG MANAGER provides two “fast path” methods for accessing the options panels:

- From the Command line of any panel, type OPT number, where number is the number assigned to an options panel (Table 13), and press Enter. The options panel is displayed.

   To return to the previous panel, press END.

<table>
<thead>
<tr>
<th>Number</th>
<th>Panel name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Options</td>
</tr>
<tr>
<td>1</td>
<td>General Options</td>
</tr>
<tr>
<td>2</td>
<td>Object Use Options</td>
</tr>
<tr>
<td>3</td>
<td>JCL Generation Options</td>
</tr>
<tr>
<td>4</td>
<td>Datasets</td>
</tr>
<tr>
<td>5</td>
<td>SQL and Confirm Options</td>
</tr>
</tbody>
</table>
Defining an options data set

Chapter 3 Setting up CATALOG MANAGER

Table 13 Options panels (Part 2 of 2)

<table>
<thead>
<tr>
<th>Number</th>
<th>Panel name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>SQL Select</td>
</tr>
<tr>
<td>7</td>
<td>Colors</td>
</tr>
<tr>
<td>8</td>
<td>Switches</td>
</tr>
<tr>
<td>9</td>
<td>Describe Options</td>
</tr>
</tbody>
</table>

- From the Command line of the Options panel, type number, where number is the number assigned to an options panel (Table 13), and press Enter. The options panel is displayed.

To return to the Options panel, press END.

The following sections describe each group of options on the Options panel.

Defining an options data set

When you select an action from the Primary Menu panel after you first invoke the product, you are prompted to create a data set in which to store the settings for your user options.

1 On the Command line of the Primary Menu panel, select an action.

If you have not defined a data set for your TSO ID, the Options Dataset not defined panel is displayed (Figure 20).
2 In the **Options dataset** field, type the name of a partitioned data set.

**NOTE**

If you enclose the data set name in single quotes, ISPF does not add the prefix (or high-level qualifier) to the name. If you do not enclose the data set name in single quotes, ISPF adds the prefix (if you have defined it).

To improve system performance, type the name of a partitioned data set extended (PDSE).

3 Choose one of the following options:

- To create a new data set, type **N**.

  The Allocate Data Set panel is displayed (Figure 21). Go to step 4.
Setting basic options

The fields on the Options panel (Figure 22) have the widest effect on CATALOG MANAGER operation. The primary options on this panel are the default SQLID and the DB2 VCAT name. Usually, these options are established when CATALOG MANAGER is installed. Review the values on this panel before performing any tasks in CATALOG MANAGER.

1 From the Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

To use an existing data set, type Y.

The panel for the action that you selected is displayed.

4 Modify the values for the data set as needed.

5 In the Allocate data set with the following values field, type Y, and press Enter.

The panel for the action that you selected is displayed.
2 Press Enter.

The Options panel is displayed (Figure 22).

**Figure 22  CATALOG MANAGER Options panel**

![Options panel](image)

3 *(optional)* In the User options Dsn field, change the name of the data set and a member in which CATALOG MANAGER stores your user settings. You can use this data set to set your options once in CATALOG MANAGER and share the options in multiple environments.

The Options Dataset has changed panel is displayed (Figure 23).
Setting basic options

Chapter 3 Setting up CATALOG MANAGER 81

To create a new data set, type a name in the Options dataset field and type N.

To use an existing data set, type Y.

4 In the Describe pds member field, type the name of a member in which CATALOG MANAGER stores the overrides for the DESCRIBE option. CATALOG MANAGER stores this member in the same data set that you specify in the User options Dsn field.

NOTE Do not specify the same member name for the DESCRIBE option as you do for the user options.

You must specify a member name to modify the options on the DESCRIBE report. For information, see “Setting DESCRIBE options” on page 98.

5 In the DB2 VCAT field, type a value for the high-level qualifier of the DB2 catalog to which you are attached.

6 (optional) In the Default SQLID field, type an initial SQL ID that is different from your TSO ID. Setting an initial SQL ID does not alter the function of the SET sql ID command during a session.

Table 14 describes the values that are valid for the Default SQLID field.
Setting general options

Table 14  Valid values for the Default SQLID field

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sql ID</td>
<td>After each logon, CATALOG MANAGER sets your current SQL ID to the SQL ID in the Default SQLID field.</td>
</tr>
</tbody>
</table>
| *PERSIST | After each logon, CATALOG MANAGER sets your current SQL ID to the SQL ID that was current when you ended your last CATALOG MANAGER session.  

The SQL ID that was current when you ended your last CATALOG MANAGER session can be any of the following values:  

- an SQL ID that was set in the Default SQLID field  
- the last SQL ID that was set by using the SET sql ID command during the previous session  
- an SQL ID that was set by previous use of the *PERSIST value in the Default SQLID field |

7 In the Max Lines per list field, type the maximum number of lines to display in a list. Valid values are 0 through 9999. To improve the performance of CATALOG MANAGER when working with large catalogs, type a value that is smaller than the default of 300.

8 In the Maximum # of select lines field, type the maximum number of rows to be displayed with the SELECT command.

9 In the Profile field, type the name of a session profile.

A session profile can be used to customize the Primary Menu panel, commands table, or initial list filter for a user. For more information, see Chapter 11, “Customizing CATALOG MANAGER command access.”

10 Press END to return to the Primary Menu panel.

Setting general options

1 From the Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

2 Press Enter.

The Options panel is displayed.
In the Edit General options field, type Y.

The General Options panel is displayed (Figure 24).

Figure 24  General Options panel

<table>
<thead>
<tr>
<th>Command</th>
<th>General Options</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal point</td>
<td>Period or comma</td>
<td></td>
</tr>
<tr>
<td>SQL string delimiter</td>
<td>Quote or double-quote</td>
<td></td>
</tr>
<tr>
<td>Lines per page</td>
<td>75</td>
<td>0-199, 0 - unlimited</td>
</tr>
<tr>
<td>DB2 feedback size</td>
<td>64</td>
<td>16-999, Size of feedback area in KB</td>
</tr>
<tr>
<td>Truncation chars</td>
<td>Long name truncation character</td>
<td></td>
</tr>
<tr>
<td>Truncation position</td>
<td>L</td>
<td>L-Left, M-Middle, R-Right</td>
</tr>
<tr>
<td>Char field max width</td>
<td>30</td>
<td>10-99 For list display</td>
</tr>
<tr>
<td>Terse level</td>
<td>VERBOSE</td>
<td>TERSE/VERBOSE</td>
</tr>
</tbody>
</table>

In the Decimal point field, type a period (.) or a comma (,) to use as the decimal point.

At startup, CATALOG MANAGER reads the DSNHDECP module and sets the Decimal Point value to the character that is set in DB2, either a period or comma. You can change the option only for the current session (for example, for testing or connecting to a different SSID). CATALOG MANAGER defaults to the DB2 setting at the next startup.

In the SQL string delimiter field, type a delimited identifier.

The names of DB2 objects can be composed of ordinary identifiers or delimited identifiers. Ordinary identifiers include the letters A through Z, the digits 0 through 9, the three national characters @, #, $, and the underscore character (_). The first character cannot be a digit, and embedded blanks are not permitted. Some SQL keywords might not be ordinary identifiers.

Delimited identifiers do not follow these rules. Delimited identifiers must be enclosed in SQL escape characters, which are normally double quotation marks ("") but might be set to single quotation marks (‘’) at DB2 installation.
You can use delimited identifiers for the names of tables, views, aliases, synonyms, columns, and indexes. If you enter an object name that is delimited, CATALOG MANAGER encloses it within the appropriate SQL escape characters. The SQL escape character is specified indirectly. The character that you do not specify as the SQL string delimiter is used as the SQL escape character.

6 In the **Lines per page** field, type the maximum number of lines per page on the print data set. Valid values are 0 through 199.

7 In the **DB2 feedback size** field, type the amount of KB to provide for DB2 commands. Valid values are 16 through 999.

8 In the **Truncation chars** field, type the characters that replace the beginning and end of a truncated string in an object name that is too long to be displayed.

9 In the **Truncation position** field, type **L**, **M**, or **R** to specify the location of characters to be omitted in object names that are too long to be displayed.

<table>
<thead>
<tr>
<th>To replace characters</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>at the left end (beginning) of the name</td>
<td>L</td>
</tr>
<tr>
<td>in the middle of the name</td>
<td>M</td>
</tr>
<tr>
<td>at the right end (end) of the name</td>
<td>R</td>
</tr>
</tbody>
</table>

10 In the **Char field max width** field, type the maximum length of a character column to be displayed. Valid values are 10 through 99. If the actual width of the column name exceeds this value, CATALOG MANAGER truncates the column name accordingly.

For more information about customizing list displays, see “Customizing object list displays” on page 196.

11 In the **Terse level** field, type **TERSE** or **VERBOSE** to indicate the amount of output that you want CATALOG MANAGER to produce in the DDL process.

This option is supported only for partitioned table spaces and indexes.

To produce terse DDL, CATALOG MANAGER requires that all attributes for all partitions match the attributes for the first partition. If any of the attributes are different, CATALOG MANAGER produces verbose DDL.

12 Press END to return to the Options panel.
Setting object use options

1. From the Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

2. Press Enter.

   The Options panel is displayed.

3. In the Edit Object selections field, type Y.

   The Object Use Options panel is displayed (Figure 25).

Figure 25  Object Use Options panel

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Object Use Options 1 to 14 of 14</th>
<th>Scroll ===&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET CURRENT SQLID to . .</td>
<td>CREATEDBY CREATEDBY/CREATOR/NONE/other</td>
<td></td>
</tr>
<tr>
<td>Synonyms in Mixed list .</td>
<td>Y</td>
<td>Y/N Display synonyms</td>
</tr>
<tr>
<td>Packages in Mixed list .</td>
<td>Y</td>
<td>Y/N Display packages</td>
</tr>
<tr>
<td>Plans in Mixed list . .</td>
<td>Y</td>
<td>Y/N Display plans</td>
</tr>
<tr>
<td>Include in HDDL and HDESCRIBE</td>
<td>Include in HDDL commit counts</td>
<td></td>
</tr>
<tr>
<td>Tablespace . . . . . . .</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Table . . . . . . . . .</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Foreign key . . . . . .</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>View . . . . . . . . .</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Index . . . . . . . . .</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Synonym . . . . . . . .</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Alias . . . . . . . . .</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Plan . . . . . . . . . .</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Trigger . . . . . . . .</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Commit frequency count .</td>
<td>0-999 How often to insert commits</td>
<td></td>
</tr>
</tbody>
</table>

4. In the SET CURRENT SQLID to field, specify the value for the SET CURRENT SQLID statement that the DDL, HDDL, and MDDL commands produce for views and materialized query tables.

   Table 15 lists valid values for the SET CURRENT SQLID to field.

Table 15  Valid values for the SET CURRENT SQLID to field

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATEDBY or CREATOR</td>
<td>indicates the authorization ID in the DB2 catalog</td>
</tr>
<tr>
<td>NONE or blank</td>
<td>indicates that the statement is not included</td>
</tr>
<tr>
<td>&lt;other&gt;</td>
<td>represents a variable that indicates any value</td>
</tr>
</tbody>
</table>

5. Specify whether to display synonyms, packages, and plans in a mixed list.
Setting JCL Generation options

6 Specify whether to include objects in the CREATE object statements that the HDDL and HDESCRIBE commands produce.

If you exclude tables from the HDDL, CATALOG MANAGER also excludes materialized query tables (MQTs).

7 In the Include in the HDDL commit counts field, specify whether to generate a COMMIT statement after the number of CREATE object statements specified in the Commit frequency count field for table spaces, tables, views, and indexes.

8 For objects in which you specified Y in the Include in the HDDL commit counts field, in the Commit frequency count field, specify the number of CREATE object statements to execute before inserting a COMMIT statement. Valid values are 0 through 999.

9 Press END to return to the Options panel.

Setting JCL Generation options

CATALOG MANAGER can generate JCL and submit a job to execute utilities against objects in a list. The JCL options apply to both BMC utilities and IBM DB2 utilities. To submit utility jobs from CATALOG MANAGER successfully, you must set the JCL parameters correctly.

1 From the Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

2 Press Enter.

   The Options panel is displayed.

3 In the Edit JCL Generation options field, type Y.

   The JCL Generation Options panel is displayed (Figure 26).
4 In the **Separate steps** field, type **Y** or **N** to specify whether to generate a separate job step in the JCL for each utility statement.

5 In the **Generate using Worklist** field, type **Y**, **N**, or **E** to specify how to generate the JCL.

<table>
<thead>
<tr>
<th>To generate the utility JCL by</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>using a worklist</td>
<td>Y</td>
</tr>
<tr>
<td>not using a worklist</td>
<td>N</td>
</tr>
<tr>
<td>using a worklist and to include event information</td>
<td>E</td>
</tr>
</tbody>
</table>

If you have the BMC DASD MANAGER PLUS product installed, DASD MANAGER PLUS inserts rows into the DASD MANAGER PLUS EVENTS table.
The choices that you make for the **Separate steps** and the **Generate using Worklist** fields determine how utility statements are created:

<table>
<thead>
<tr>
<th>Choice for Separate steps</th>
<th>Choice for Generate using Worklist</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Y or E</td>
<td>CATALOG MANAGER generates JCL by using a worklist. A single WORKLIST DD statement in the JCL includes all of the utility worklist commands.</td>
</tr>
<tr>
<td>N</td>
<td></td>
<td>CATALOG MANAGER generates JCL with separate job steps for each utility command. Each SYSIN DD statement in the JCL includes a single utility command.</td>
</tr>
<tr>
<td>N</td>
<td>Y or E</td>
<td>CATALOG MANAGER generates JCL by using a worklist. A single WORKLIST DD statement in the JCL includes all of the utility worklist commands.</td>
</tr>
<tr>
<td>N</td>
<td></td>
<td>CATALOG MANAGER generates JCL with a single job step for the utility commands. The SYSIN DD statement in the JCL includes all of the utility commands.</td>
</tr>
</tbody>
</table>

**6** In the **Object limit** field, type the maximum number of objects to be included in the JCL. Valid values are 0 through 9999. A value of 0 indicates that an unlimited number of objects can be included in the JCL.

**7** In the **Group attach** field, type an SSID for the DB2 group attachment name.

**8** In the **Edit JCLGen options** field, type **Y** to display a series of panels for setting options for the product options file. For more information, see “Setting product options” on page 99.

**9** Press END to return to the Options panel.

### Setting data set options

The Datasets panel enables you to specify default data set names (DSNs) that are needed by CATALOG MANAGER to perform some of its functions. The defaults specified on the Datasets panel are used to access the associated data sets as follows:

- If the data set name is enclosed in single quotation marks, the name is used as is.
- If the data set name contains any ISPF variables (such as &ZPREFIX), the variables are substituted before the data set is allocated.
If the data set name has no apostrophes and &ZPREFIX contains a value, the &ZPREFIX value is prefixed as in the first node of the name.

**To set data set options**

1. From the Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

2. Press Enter.

   The Options panel is displayed.

3. In the **Edit Dataset names** field, type Y.

   The Datasets panel is displayed (**Figure 27**).

**Figure 27  Datasets panel**

<table>
<thead>
<tr>
<th>Datasets</th>
<th>1 to 11 of 11</th>
<th>Scroll ===&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print Dsn. . . . . . . . . . . .</td>
<td>&amp;ZUSER..BMCCAT.PRINT</td>
<td></td>
</tr>
<tr>
<td>Catalog Manager Work Dsn . . .</td>
<td>&amp;ZUSER..BMCCAT.WORK</td>
<td></td>
</tr>
<tr>
<td>SQL Output Dsn . . . . . . . . .</td>
<td>&amp;ZUSER..BMCCAT.SQL</td>
<td></td>
</tr>
<tr>
<td>Online Bind default DBRM Dsn . .</td>
<td>'BMCACT.V71S2.DBRM'</td>
<td></td>
</tr>
<tr>
<td>User Utilities Profile Dsn . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JCL Generation Dsn . . . . . . .</td>
<td>&amp;ZUSER..BMCCAT.JCL()</td>
<td></td>
</tr>
</tbody>
</table>
Setting SQL and confirm options

4 In the Print Dsn field, type the name of the print output data set. CATALOG MANAGER dynamically allocates the sequential data set the first time that it is used.

The PRINT command opens the data set for output when you first issue the command in each session. Additional print output is appended to the data set until you issue one of the following commands:

- HC to submit the data set for printing
- PRINT CLOSE to close and deallocate the data set

5 In the Catalog Manager Work Dsn field, type the name of the work data set in which statements that the HDDL command produces are stored. CATALOG MANAGER dynamically allocates the sequential data set the first time that it is used.

6 In the SQL Output Dsn field, type the name of the SQL output data set. CATALOG MANAGER dynamically allocates the sequential data set the first time that it is used.

7 In the Online Bind default DBRM Dsn field, type the name of the data set that is used in a BIND command.

8 In the User Utilities Profile Dsn field, type the preallocated data set name and a member name, if the data set is partitioned.

9 In the JCL Generation Dsn field, type the default name of the partitioned data set that is used for utilities, the HC command, worklist job generation, and other jobs in which CATALOG MANAGER builds JCL as needed.

10 Press END to return to the Options panel.

Setting SQL and confirm options

On the SQL and Confirm Options panel, you can specify the default settings of options that are used by SQL commands, as well as the options that are displayed on Confirm SQL panels. Confirm SQL panels are found near the end of many CATALOG MANAGER processes. They display a summary of the commands to be written to the SQL statement that will process the desired actions. Set the values that you expect to use most frequently for all SQL processing to avoid having to set them every time that you execute an SQL command.

NOTE
CATALOG MANAGER uses 2 MB for the memory size for the work area that CATALOG MANAGER uses for processing SQL.
1. From the Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

2. Press Enter.

   The Options panel is displayed.

3. In the Edit SQL and Confirm options field, type Y.

   The SQL and Confirm Options panel is displayed (Figure 28).

   **Figure 28 SQL and Confirm Options panel**

<table>
<thead>
<tr>
<th>Command ====&gt;</th>
<th>SQL and Confirm Options 1 to 16 of 16</th>
<th>Scroll ====&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL processing (INSERT,DELETE,UPDATE,CREATE and executed from the SQL Table)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auto-commit . . . . . . Y N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate SQL/DSN . . . . N Y/N</td>
<td>Default value for Generate/Create</td>
<td></td>
</tr>
<tr>
<td>Confirm SQL and DSN panel defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit . . . . . . . . . N Y/N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save in SQL table . . . . N A/Y A/Y-Append, R-Replace, N-No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute . . . . . . . . N Y/N Drop/Revoke always default to N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last used is default . . N Y/N Applies to the 3 preceding options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drop recovery and revoke reassign defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add dependency list . . . N Y/N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drop recovery on . . . . N Y/N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log Image copies . . . . N Y/N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add cascade report . . . N Y/N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plan table . . . . . . PLAN_TABLE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Specify your default options for processing SQL:

   **A** In the Auto-commit field, type Y or N to specify whether to execute a COMMIT statement after successfully executing an SQL CREATE, DELETE, INSERT, or UPDATE statement from the SQL_Table.

      The Auto-commit option does not apply to SQL SELECT statements. An SQL SELECT statement automatically forces a COMMIT statement.

   **B** In the Generate SQL/DSN field, type Y or N to specify whether the default value for the Generate SQL field is Y or N on the Create or Generate panels.
5 Specify your default options for the confirmation panels.

A In the **Edit** field, type **Y** or **N** to specify whether the default value for the **Edit** field is **Y** or **N**.

B In the **Save in SQL table** field, type **A**, **Y**, **R**, or **N** to specify the default value for the **Save in SQL table** field on a Confirm panel.

<table>
<thead>
<tr>
<th>To default to this action</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>append the SQL to the SQL in the SQL_Table</td>
<td>A</td>
</tr>
<tr>
<td>save the SQL in the SQL_Table</td>
<td>Y</td>
</tr>
<tr>
<td>replace the SQL in the SQL_Table</td>
<td>R</td>
</tr>
<tr>
<td>discard the SQL</td>
<td>N</td>
</tr>
</tbody>
</table>

C In the **Execute** field, type **Y** or **N** to specify the default value for the **Execute SQL** field.

D In the **Last used is default** field, type **Y** or **N** to specify whether to override the values specified for the **Edit**, **Save in SQL table**, and **Execute** fields on this panel with the values that were last used on a Confirm panel.

6 Specify your default options for drop recovery and revoking privileges.

A In the **Add dependency list** field, type **Y** or **N** to specify whether to see the list of dependent objects that will be dropped with an object.

B In the **Drop recovery on** field, type **Y** or **N** to specify whether to be able to recover an object and its dependent structure definitions.

C In the **Log image copies** field, type **Y** or **N** to specify whether to be able to recover a copy of the data for an object and its dependent objects.

D In the **Add cascade report** field, type **Y** or **N** to specify whether to generate a cascade report.

7 In the **Plan table** field, type the name of the PLAN_TABLE that should be used for Explain information.

8 Press END to return to the Options panel.
Setting SQL SELECT options

Use the SQL Select panel to specify the data set name for SQL output and to set maximum values for output from the SELECT command.

1. From the Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

2. Press Enter.

   The Options panel is displayed.

3. In the Edit SQL Select options field, type Y.

   The SQL Select panel is displayed (Figure 29).

   **Figure 29   SQL Select panel**

   ![SQL Select panel]

   - Command ===> SQL Select                          1 to 6 of 6
     - Browse SQL output dataset. . . Y
     - Maximum output line length . . . 256
     - Maximum numeric field width . . . 10
     - Maximum char field width . . . 64
     - Maximum varchar field width . . . 64
     - Y/N Y-Automatically browse dataset
     - 1-4092

4. In the Browse SQL output dataset field, type Y or N to specify whether to automatically browse the SQL output data set after executing SQL.

5. In the Maximum output line length field, type the maximum number of characters in a single line of output. Valid values are 1 through 4092. The value that you specify must be smaller than the LRECL of the BMCCAT.SQL data set.

6. In the Maximum numeric field width field, type the maximum number of digits in a numeric field.
Setting panel graphic options

In the Maximum char field width field, type the maximum number of characters in a character field.

In the Maximum varchar field width field, type the maximum number of characters in a VARCHAR field.

Press END to return to the Options panel.

Setting panel graphic options

On the Colors panel you can change the Common User Access (CUA) default colors and select the type of highlighting (underscore, reverse video, or none) that identifies the currently selected field on a panel.

NOTE
Your highlighting and color capabilities depend on the type of terminal that you are using.

1 From the Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

2 Press Enter.

The Options panel is displayed.

3 In the Edit Color options field, type Y.

The Colors panel is displayed (Figure 30).
For the various colors, type the default color.

In the Highlight field, type REVERSE or USCORE to indicate the type of highlighting for a field. You can also leave the field blank and not use highlighting.

In the Set DASD MANAGER Graphics field, type Y or N to edit the graphics options that are used by the DASD MANAGER PLUS product. If you type Y and have the DASD MANAGER PLUS product installed, the DASD MANAGER PLUS Graphics Options panel is displayed. Press END to return to the Colors panel.

Press END to return to the Options panel.

**Setting CATALOG MANAGER switches**

CATALOG MANAGER switches control whether to enable various features. You can set switches in one of two ways:

- From the Command line of the Primary Menu panel or an object list panel, issue one of the following commands:
  
  — **SET switch ON**
  — **SET switch OFF**
Setting the switch is only temporary for the session. When you start a new CATALOG MANAGER session, the switch is reset.

- From the Switches panel, type over the existing value with Y or N.

To access the Switches panel, follow the steps in “To set switches” on page 96.

**To set switches**

1. From the Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

2. Press Enter.

   The Options panel is displayed.

3. In the Edit Switches field, type Y.

   The Switches panel is displayed (Figure 31).

**Figure 31 Switches panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Switches</th>
<th>Scroll ====&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>1 to 6 of 6</td>
<td></td>
</tr>
<tr>
<td>Show Server</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>DBCS</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>HDDL Auths</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>WId</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>DEBUG</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Shared data</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Caps</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Cmp &gt; 32k</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Drop</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>SQL flow</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Define No.</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Last used prof</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Revoke BY</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Server SSID</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>SQL Ownerid</td>
<td>.     .     .     .     .     .     .     .   SQL</td>
<td>SQL/TSO</td>
</tr>
<tr>
<td>Build SQLID</td>
<td>before GRANT       .     .     .     .     .   N</td>
<td>Y/N</td>
</tr>
<tr>
<td>Dev debug</td>
<td>.     .     .     .     .     .</td>
<td></td>
</tr>
</tbody>
</table>

- Build SQLID before GRANT: N
- Y/N
4 To change the value of a switch, type Y or N over the existing value.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Server</td>
<td>display the current server in the upper right corner of a panel</td>
</tr>
<tr>
<td>DEBUG</td>
<td>display dynamic SQL before executing a statement</td>
</tr>
<tr>
<td>Cmp &gt; 32k</td>
<td>compress HDDL SQL statements that are longer than 32 KB</td>
</tr>
<tr>
<td>Last used prof</td>
<td>use the last used utility profile as the default profile</td>
</tr>
<tr>
<td>DBCS</td>
<td>delimit DBCS data</td>
</tr>
<tr>
<td>Label</td>
<td>use labels as the column headings when browsing SQL</td>
</tr>
<tr>
<td>Drop</td>
<td>execute SQL DROP statements</td>
</tr>
<tr>
<td>Revoke BY</td>
<td>generate the BY clause of a REVOKE statement</td>
</tr>
<tr>
<td></td>
<td>If Y, the BY clause will always be generated. If N, the BY clause will be generated only if the current SQLID has SYSADM or SYSCTRL authority. If the current SQLID does not have SYSADM or SYSCTRL authority and is not the grantor, the product issues an error message.</td>
</tr>
<tr>
<td>HDDL Auths</td>
<td>include GRANT statements in HDDL output</td>
</tr>
<tr>
<td>Shared data</td>
<td>for a data sharing environment, send the group attachment name (SSID) to the JCL Generation component to generate utility jobs</td>
</tr>
<tr>
<td>SQL flow</td>
<td>maintain original column alignment that existed in the SQL statements before substituting host variables</td>
</tr>
<tr>
<td>Server SSID</td>
<td>display the first four characters of the server (instead of the SSID) in the upper left corner</td>
</tr>
<tr>
<td>_ Wild</td>
<td>for table lists, consider an underscore as a wildcard character if no other wildcard characters (such as % or *) are included in the qualifier</td>
</tr>
<tr>
<td>Caps</td>
<td>translate characters in panel fields that are used to enter object names to uppercase characters</td>
</tr>
<tr>
<td>Build SQLID before GRANT</td>
<td>generate a SET CURRENT SQLID = grantor statement before each GRANT statement</td>
</tr>
<tr>
<td></td>
<td>CATALOG MANAGER generates these GRANT statements with the HGRANT and HDDL commands, and the REVOKE command with the Reassign Grants option.</td>
</tr>
</tbody>
</table>
5 In the Define No field, type Y, N, or C to indicate how to include the DEFINE parameter in DDL or HDDL for a table space or index.

<table>
<thead>
<tr>
<th>If you want to</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>include the DEFINE NO parameter</td>
<td>Y</td>
</tr>
<tr>
<td>include the DEFINE YES parameter</td>
<td>N</td>
</tr>
<tr>
<td>generate objects based on their status in the DB2 catalog:</td>
<td>C</td>
</tr>
<tr>
<td>If the SPACE attribute in the SYSIBM.SYSTABLEPART table equals -1, include the DEFINE NO parameter.</td>
<td></td>
</tr>
<tr>
<td>If the SPACE attribute in the SYSIBM.SYSTABLEPART table is greater than or is equal to 0, include the DEFINE YES parameter.</td>
<td></td>
</tr>
</tbody>
</table>

6 In the SQL Ownerid field, type TSO or SQL to indicate whether the owner of the default SQL_Table member is the user’s TSO ID or SQLID.

**NOTE**

Do not enter a value in the Dev debug field. This field is for use by BMC Customer Support only.

7 Press END to return to the Options panel.

**Setting DESCRIBE options**

The DESCRIBE command displays detailed information that is stored in the DB2 catalog about a specific object, including the object’s structure and dependencies. The DESCRIBE command options control how this detailed information is presented for plans, packages, tables, and views. For more information about the DESCRIBE command, see “Describing list objects” on page 201.

1 From the Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

2 Press Enter.

The Options panel is displayed.

3 In the Edit Describe options field, type Y.

The Describe Options panel is displayed (Figure 32).
Setting product options

You can specify many options for generating the JCL for individual data sets. Many of these options were formerly available only through modifications to the skeleton libraries (SLIBs). Over 300 parameters are available from the options panels. These parameters are initialized from an extension to the installation options module.

4 In the **Output format** field, type **Edit**, **Browse**, or **Enhanced** to indicate how to present report data.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit</td>
<td>enables you to edit the report in a data set</td>
</tr>
<tr>
<td>Browse</td>
<td>enables you to browse the report in a data set</td>
</tr>
<tr>
<td>Enhanced</td>
<td><em>(default)</em> provides the report in color and enables you to use the GET command to analyze statements within a package</td>
</tr>
</tbody>
</table>

5 In the **Output dsn** field, type the name of the data set in which CATALOG MANAGER writes the DESCRIBE report.

6 Press END to return to the Options panel.
Setting product options

1. From the CATALOG MANAGER Primary Menu panel, an object list panel, or a utility panel, on the Command line, type OPTIONS (OPT).

2. Press Enter.

   The Options panel is displayed.

3. In the Edit JCL Generation options field, type Y. Press Enter.

   The JCL Generation Options panel is displayed.

4. In the Edit JCLgen options field, type Y. Press Enter.

   The JCL Generation Update - Main Menu panel is displayed (Figure 33).

   **Figure 33 JCL Generation Update Panel**

   ![JCL Generation Update Panel](image)

   Select an option number and press Enter.

   1. Jobcard Options
   2. Steplib Options
   3. Static Data Set Options
   4. Tape Options
   5. Individual Data Set Options
   6. Generation Data Group Options (GDGs)
   7. Debugging, Display and Execution Options
   8. Utility Options
   9. Listdef and Template Data Sets
   10. Not Applicable
   11. User Defined Variable Values
   12. Product Options File (POF) Functions

5. Type an option number, and then press Enter to display the panel for the option that you want to set.

<table>
<thead>
<tr>
<th>Option</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobcard Options</td>
<td>“Setting the JCL options for job cards” on page 101</td>
</tr>
<tr>
<td>Steplib Options</td>
<td>“Setting the JCL options for STEPLIBs” on page 104</td>
</tr>
<tr>
<td>Static Data Set Options</td>
<td>“Setting the JCL options for static data sets” on page 105</td>
</tr>
<tr>
<td>Tape Options</td>
<td>“Setting the JCL options for tapes” on page 109</td>
</tr>
<tr>
<td>Individual Data Set Options</td>
<td>“Setting the JCL options for temporary work data sets” on page 112</td>
</tr>
<tr>
<td></td>
<td>“Setting the JCL options for permanent data sets” on page 114</td>
</tr>
</tbody>
</table>
To navigate the JCL Generation Options panels, use the following guidelines:

- To view all of the panels in sequence, press Enter.

- To save your settings and return to the JCL Generation Update - Main Menu panel, press END at any panel in the sequence.

- To display the JCL Generation Options panel from the JCL Generation Update - Main Menu panel, press END.

### Setting the JCL options for job cards

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Jobcard Options panel to specify information about the job cards used in the JCL.
Setting the JCL options for job cards

To set the JCL options for job cards

1 Use the following menu selections to display the JCL Generation Jobcard Options Update panel (Figure 34):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Jobcard Options</td>
</tr>
</tbody>
</table>

Figure 34  JCL Generation Jobcard Options Update panel

AJXOJOBP ---------- JCL GENERATION JOBCARD OPTIONS UPDATE ------------------
COMMAND ===> 
Type data and press Enter.

Is a TSO submit exit used to generate jobcards? . . N  (Y/N)
Enter Jobcards below:
//&USERID.&JOBCHAR JOB (&ZACCTNUM),"&PGMR",
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
// NOTIFY=&USERID
/*ROUTE XEQ BMCPLX1
/*JOBPARM SYSAFF=&ZSYSID
Jcllib . . . . . .
Sysexec. . . . . . (See JCL Reference for valid options)
Region size . . . 0M (See JCL Reference for valid options)
Memlimit . . . . NOLIMIT (See JCL Reference for valid options)
Time parameter . . (See JCL Reference for valid options)
System MLIB. . . . ISP.SISPMENU
Runtime HLQ. . . . AEX.QA101
LLQ. . . . . . . . BMC
ULLQ . . . . . . . (Leave blank if not using runtime enablement)

2 In the Is a TSO submit exit used to generate jobcards? field, type Y or N.

3 At Enter Jobcards below, type the job statement information that you want to add to the JCL.

4 In the Jcllib field, enter the name of one of the following types of partitioned data sets (PDSs):
   ■ a PDS that contains customized JCL to be included in the job
   ■ a PDS that specifies the cataloged procedures (PROCs) that are used for non-worklist JCL
5 In the **Sysexec** field, type the name of the PDS in which a REXX EXEC is a member.

---

**TIP**

To indicate the data set name for a different SSID, append the &SSID or &MSSID symbolic variable to the name.

---

6 In the **Region size** field, type the amount of memory to allocate for each step so that your job can run.

7 In the **Memlimit** field, type the limit on the above-the-bar memory for an address space.

8 In the **Time parameter** field, type the value for the TIME limit for each step in a batch job stream.

9 In the **System MLIB** field, type the name of the system ISPF message library.

10 In the **Runtime HLQ** field, type the high-level qualifier (HLQ) for ISPF data sets for the installation environment.

   During installation, if you chose to use the runtime enablement (RTE) feature, the Installation System set this value to an HLQ for user runtime libraries. If you chose not to use RTE, the Installation System set the value to an HLQ for Execution.

   This value supports the following symbolic variables:

   - &DB2V2 and &DB2V3, which resolve to the version of DB2
   - &SSID, which resolves to the DB2 subsystem ID

   When you include the &SSID symbolic variable, the product can use a single POF with multiple subsystems.

11 In the **LLQ** field, type the low-level qualifier for ISPF data sets for the installation environment.

   During installation, if you chose to use the runtime enablement feature, the Installation System set this value to BMC. If you chose not to use the feature, the Installation System set the value to DB.

12 In the **ULLQ** field, type the low-level qualifier for user-defined data sets for the installation environment.

   During installation, if you chose to use the runtime enablement feature, the Installation System set this value to blank, and you should not change the value. If you chose not to use the feature, the Installation System set the value to UDB.
Press END to save your changes and return to the JCL Generation Update - Main Menu panel.

**Setting the JCL options for STEPLIBs**

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the STEPLIB Options panel to specify the load libraries that appear in jobs that CATALOG MANAGER creates.

1. Use the following menu selections to display the JCL Generation STEPLIB Options Update panel (Figure 35):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Steplib Options</td>
</tr>
</tbody>
</table>

**Figure 35  JCL Generation STEPLIB Options Update panel**

```
COMMAND ===> JCL GENERATION STEPLIB OPTIONS UPDATE
Type data and press Enter.
DSNEXIT . . . . . . . SYS3.DBDC.DSNEXIT
DB2 DSNLOAD . . . . . . SYS2.DB2V10M.DSNLOAD
Override lib. . . . . .
ALTER/CHANGE MANAGER. .
DASD MANAGER PLUS . .
EXECUTION . . . . . . . ’BMCADMN.V101.10.BMCLINK’
COPY PLUS . . . . . .
REORG PLUS . . . . . .
LOADPLUS . . . . . .
UNLOAD PLUS . . . . .
RECOVER PLUS . . . .
CHECK PLUS . . . . .
SQL EXPLORER . . . .
Additional lib . . .
IOA LOAD 1 . . . . .
IOA LOAD 2 . . . . .
```
2 Specify the data set names for the libraries:

**TIP**
To indicate the data set name for a different SSID, append the &SSID or &MSSID symbolic variable to the name.

**A** In the **DSNEXIT** field, type the data set name of the DB2 EXIT (DSNEXIT) library.

**TIP**
To indicate the version of DB2, append the &DB2V2 or &DB2V3 symbolic variable to the name.

**B** In the **DB2 DSNLOAD** field, type the data set name of the library in which the DB2 DSN Command Processor load modules are stored.

**C** In the **Override lib** field, type the data set name for the override LINK library that should appear first in the STEPLIB statement.

**D** Type the data set names of the LINK libraries for the BMC products.

**E** In the **Additional lib** field, type the data set name for the additional LINK library that should appear last in the STEPLIB statement.

**F** *(Database Performance for DB2 solution only)* In the **IOA LOAD 1** and **IOA LOAD 2** fields, type the names of the LINK libraries that are used for automating the CONTROL-M jobs.

3 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.

---

**Setting the JCL options for static data sets**

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Static Data Set Options panel to specify the options for sizing and cleaning up your data sets.
To set the JCL options for static data sets

1 Use the following menu selections to display the JCL Generation Static Data Set Options Update panel (Figure 36):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Static Data Set Options</td>
</tr>
</tbody>
</table>

Figure 36 JCL Generation Static Data Set Options panel

```
--------------- JCL GENERATION STATIC DATA SET OPTIONS UPDATE -------------------
COMMAND ===>  
Type data and press Enter.
Data set sizing option... N   (N-No Sizing,B-Bmcstats,
                              C-DB2 Catalog,O-Object Sampling)
Data set sizing device... 3390 (3380/3390)
Max cylinders........... 99999 (Do not exceed this primary value in JCL.)
If max cylinders are exceeded, use the following for DASD data sets
  Max primary quantity.... 10  (Cylinders, 1 - 9999)
  Max secondary quantity... 2  (Cylinders, 1 - 9999)
  Max unit count..........        (Blank or 1 - 59 volumes)
Include data set cleanup step... N  (Y/N)
Return code for cleanup step... 4  (04)
Temporary unit............. SYSDA  (SYSDA, SYSALLDA, etc.)

Note See Debugging, Display and Execution Options to display sizing options in the JCL.
```

2 In the Data set sizing option field, type N, B, C, or O to specify the sizing method, as shown in Table 16.

**NOTE**

Whether or not data set sizing is performed, DB2 catalog access is required to resolve any symbolic variables. For more information about data set sizing, see “Sizing JCL Generation data sets” on page 363.
You can specify whether to include comments in the Execution JCL that show statistics for determining data set sizes. For more information, see “Setting the JCL debugging, display, and Execution options” on page 121.

3 In the Data set sizing device field, specify the type of DASD to use in calculating the sizes of data sets.

4 In the Max cylinders field, specify the maximum number of cylinders for a data set.

5 Specify the values that should be used for the following options when the maximum number of cylinders are exceeded:

A In the Max primary quantity field, type the number of cylinders for the maximum primary quantity.

B In the Max secondary quantity field, type the number of cylinders for the maximum secondary quantity.

C In the Max unit count field, type the maximum number of volumes.

### Table 16 Data set sizing options

<table>
<thead>
<tr>
<th>For this method</th>
<th>Type</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>no data set sizing</td>
<td>N</td>
<td>The product uses the default primary and secondary quantities for the permanent data sets. To modify the quantities for the primary and secondary space for a permanent data set, see step 4 on page 119.</td>
</tr>
</tbody>
</table>
| sizing by using statistics from the DASD MANAGER PLUS tables | B    | - The DASD MANAGER PLUS product must be installed and interacting with ALTER or CHANGE MANAGER to use this option.  
  - If a column is defined as a LONG VARCHAR, this option averages the row size.  
  - Current BMCSTATS statistics should be available for objects in the worklist before you select this option. |
| sizing by using statistics from the DB2 catalog | C    | - If a column is defined as a LONG VARCHAR, this option does not average the row size.  
  - Current IBM RUNSTATS catalog statistics should be available for objects in the worklist before you select this option. |
| estimating sizes based on physical, random sampling of VSAM data sets | O    | - If a column is defined as a LONG VARCHAR, this option averages the row size.  
  - You can use this option if the statistical information in the DB2 catalog or in the DASD MANAGER PLUS tables is not current. However, JCL generation might take additional time to complete. |

- You can specify whether to include comments in the Execution JCL that show statistics for determining data set sizes. For more information, see “Setting the JCL debugging, display, and Execution options” on page 121.

3 In the Data set sizing device field, specify the type of DASD to use in calculating the sizes of data sets.

4 In the Max cylinders field, specify the maximum number of cylinders for a data set.

5 Specify the values that should be used for the following options when the maximum number of cylinders are exceeded:

A In the Max primary quantity field, type the number of cylinders for the maximum primary quantity.

B In the Max secondary quantity field, type the number of cylinders for the maximum secondary quantity.

C In the Max unit count field, type the maximum number of volumes.
If you want to use multiple data sets on DASD, specify a value greater than 1 for **Max unit count**. On the JCL Generation Individual Data Set Options Update panels (see page 114), specify the name of a DASD unit.

In the **Include data set cleanup step** field, type **Y** or **N** to specify whether to generate a step in the JCL to delete the permanent work data sets.

You can generate the JCL for a job step that automatically deletes many of the permanent work data sets that Execution creates. Execution creates these data sets during worklist processing and sets the disposition of the data sets to new, catalog, catalog (DISP=(NEW,CATLG,CATLG)).

The cleanup job step is performed only if the condition code returned from any previous job step is less than or equal to the number that is specified in the **Return code for cleanup step** field. Table 17 lists the types of work data sets that are included in the cleanup job step. These data sets are automatically deleted unless otherwise noted.

### Table 17  Work data sets in the JCL cleanup job step

<table>
<thead>
<tr>
<th>Work data set</th>
<th>dname</th>
<th>Used in JCL cleanup by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discard (SYSDISC)&lt;sup&gt;a&lt;/sup&gt;</td>
<td>SYSDnnnn</td>
<td>BMC LOADPLUS IBM LOAD</td>
</tr>
<tr>
<td>Error</td>
<td>SYSERnnn</td>
<td>BMC CHECK PLUS BMC LOADPLUS IBM CHECK DATA IBM LOAD</td>
</tr>
<tr>
<td>Map</td>
<td>SYSMAP</td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Punch&lt;sup&gt;a&lt;/sup&gt;</td>
<td>SYSPUNCH</td>
<td>BMC REORG PLUS IBM REORG</td>
</tr>
<tr>
<td>Unload (SYSREC)&lt;sup&gt;b&lt;/sup&gt;</td>
<td>SYSRnnnn or Rnnnnyyy</td>
<td>BMC REORG PLUS IBM REORG The unload data sets that are used by REORG PLUS and IBM REORG are deleted automatically.</td>
</tr>
<tr>
<td>Work</td>
<td>not applicable</td>
<td>utilities that are listed in Table 18</td>
</tr>
</tbody>
</table>

<sup>a</sup> These data sets are specified in the cleanup job step but are commented out. You must edit the Execution JCL and remove the comment delimiters to delete these data sets automatically.

<sup>b</sup> Other unload data sets that are used by the BMC LOADPLUS and UNLOAD PLUS utilities and IBM LOAD utility are specified in the cleanup job step but are commented out. You must edit the Execution JCL and remove the comment delimiters to delete the other unload data sets automatically.
Table 18 lists the work data sets that are used by the corresponding utilities.

### Table 18 Work data sets used by utilities

<table>
<thead>
<tr>
<th>Work data set</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORTOUT</td>
<td>BMC CHECK PLUS</td>
</tr>
<tr>
<td>SORTPnnn</td>
<td>BMC LOADPLUS</td>
</tr>
<tr>
<td>SORTOonnn</td>
<td>BMC REORG PLUS</td>
</tr>
<tr>
<td></td>
<td>IBM CHECK DATA</td>
</tr>
<tr>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td></td>
<td>IBM REORG</td>
</tr>
<tr>
<td>SYSUTnnn</td>
<td>BMC CHECK PLUS</td>
</tr>
<tr>
<td>SUTnnn</td>
<td>BMC LOADPLUS</td>
</tr>
<tr>
<td>WRKnnn</td>
<td>BMC REORG PLUS</td>
</tr>
<tr>
<td></td>
<td>BMC RECOVER PLUS</td>
</tr>
<tr>
<td></td>
<td>IBM CHECK DATA</td>
</tr>
<tr>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td></td>
<td>IBM REORG</td>
</tr>
<tr>
<td></td>
<td>IBM RECOVER INDEX</td>
</tr>
<tr>
<td></td>
<td>IBM REBUILD INDEX</td>
</tr>
</tbody>
</table>

7 In the **Return code for cleanup step** field, specify the value that should be compared against the condition code that is returned from any previous job step.

You can specify any two-digit value for the return code; however, BMC recommends that you specify 4.

8 In the **Temporary unit** field, type the name of the unit that is used to allocate temporary files when JCL is generated.

9 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.

### Setting the JCL options for tapes

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Tape Options panel to specify information about tape units and stacking options.
To set the JCL options for tapes

1 Use the following menu selections to display the JCL Generation Tape Options Update panel (Figure 37):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Tape Options</td>
</tr>
</tbody>
</table>

Figure 37 JCL Generation Tape Options Update panel

```
------------------------ JCL GENERATION TAPE OPTIONS UPDATE ------------------------
COMMAND =>

Type data and press Enter.

Tape Unit 1 . . . . . . . . . CART (blank or TAPE, CART, etc)
Tape Unit 2 . . . . . . . . . TAPE (blank or TAPE, CART, etc)
Tape Unit 3 . . . . . . . . . TAPE (blank or TAPE, CART, etc)
Tape Volume count . . . . . 99 (0 - 255)
Tape EXPDT . . . . . . . . . (blank or YYYY/DDD or YYYY/DDD)
Tape RETPD . . . . . . . . . (blank or 0 - 9999 days)
Tape TRTCH . . . . . . . . . (blank or C,E,T,ET,COMP,NOCOMP)

Stacking Options
Local Primary Copy . . N (Y/N) Local Backup Copy . . N (Y/N)
Recovery Primary Copy . N (Y/N) Recovery Backup Copy . N (Y/N)
Primary Sysrec . . . . N (Y/N) Backup Sysrec . . . . N (Y/N)
Baseline Recovery . . N (Y/N) Archive . . . . . . . N (Y/N)
(CHANGE MANAGER only)
```

2 In the Tape Unit 1, Tape Unit 2, and Tape Unit 3 fields, type the names of valid tape units for your installation.

3 In the Tape Volume count field, type the maximum number of tape volumes.

4 In the Tape EXPDT field, type the expiration date for a tape.

5 In the Tape RETPD field, type the retention date for a tape.

6 In the Tape TRTCH field, type the parity, data conversion, translation, and compression value for 7-track tape drives as shown in Table 19.
For each type of copy or product data set, type Y or N to specify whether the data sets should be stacked on a tape with data sets of the same type.

Consider the following items if you choose tape stacking:

- Tape stacking options for baseline recovery data sets apply to CHANGE MANAGER only.
- Tape stacking options for backup SYSREC and archive data sets apply to CATALOG MANAGER only.
- Tape stacking is not applicable if you choose to dynamically allocate copy or unload data sets, unless you are using the BMC COPY PLUS utility to copy explicitly created table spaces.
- Tape stacking is not applicable if you use the Database Administration solution to execute a worklist in parallel.
- Tape stacking is disabled in the JCL for a worklist if all of the following conditions exist:
  - You use tape for unload (SYSREC) data sets, or you use tape because the maximum threshold value for DASD for a data set is exceeded.
  - You use tape stacking for unload (SYSREC) data sets.
  - You use the UNLOAD PLUS and LOADPLUS utilities and you include partitioned table spaces in the scope.

If you are using the UNLOAD PLUS and LOADPLUS utilities to multitask the unloading and loading of data, you cannot use tape stacking for unload (SYSREC) data sets. If you run the JCL, you might exceed the number of tape drives at your site.

### Table 19  Values for 7-track tape drives

<table>
<thead>
<tr>
<th>To choose</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>not to use seven-track tape drives</td>
<td>(leave the field blank)</td>
</tr>
<tr>
<td>odd parity, conversion, and no translation</td>
<td>C</td>
</tr>
<tr>
<td>even parity, no conversion, and no translation</td>
<td>E</td>
</tr>
<tr>
<td>odd parity, no conversion, and translation</td>
<td>T</td>
</tr>
<tr>
<td>even parity, no conversion, and translation</td>
<td>ET</td>
</tr>
<tr>
<td>data compression</td>
<td>COMP</td>
</tr>
<tr>
<td>no data compression</td>
<td>NOCOMP</td>
</tr>
</tbody>
</table>

7 For each type of copy or product data set, type Y or N to specify whether the data sets should be stacked on a tape with data sets of the same type.
8 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.

Setting the JCL options for temporary work data sets

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Options For Sort Files panel to specify information about the temporary work data sets:

- SORTWORK (SORTWK)
- DATAWORK (DATAWK)
- LOGSORT (LOGSWK)

The temporary work data sets (such as SORTWORK) are defined by using DISP=(,PASS) in the JCL.

To set the JCL options for temporary work data sets

1 Use the following menu selections to display the JCL Generation Options For Sort Files Update panel (Figure 38):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Individual Data Set Options</td>
</tr>
<tr>
<td>JCL Generation Individual Data Set Options Update</td>
<td>Sortwork</td>
</tr>
</tbody>
</table>
Figure 38  JCL Generation Options For Sort Files Update panel

| COMMAND ===>|
|----------------- JCL GENERATION OPTIONS FOR SORT FILES UPDATE -----------------|
| Type data and press Enter.|
| Number of SORTWORK Data Sets . . 5 (1 - 32) |
| SORTWORK unit name . . . . . . SYSDA (SYSDA, 3380, etc) |
| Number of DATAWORK Data Sets . . 5 (1 - 32) |
| DATAWORK unit name . . . . . . SYSDA (SYSDA, 3380, etc) |
| Number of LOGSORT Data Sets . . 1 (1 - 32) |
| LOGSORT unit name . . . . . . SYSDA (SYSDA, 3380, etc) |
| Default Primary Quantity . . 10 (Cylinders) |
| Default Secondary Quantity . . 2 (Cylinders) |
| SMS Data Class . . . . . . . (Blank or Data Class Name) |
| SMS Storage Class . . . . . . (Blank or Storage Class Name) |
| SMS Management Class . . . . (Blank or Management Class Name) |
| $SORTPARM data set name (below) |

2 Specify the options for SORTWORK data sets:

A In the Number of SORTWORK Data Sets field, type the number of SORTWORK data sets.

B In the SORTWORK unit name field, type the name of the unit for SORTWORK data sets.

3 Specify the options for DATAWORK data sets:

A In the Number of DATAWORK Data Sets field, type the number of DATAWORK data sets.

B In the DATAWORK unit name field, type the name of the unit for DATAWORK data sets.

4 Specify the options for LOGSORT data sets:

A In the Number of LOGSORT Data Sets field, type the number of LOGSORT data sets.

B In the LOGSORT unit name field, type the name of the unit for LOGSORT data sets.
5 If you typed N in the Data set sizing option field in step 2 on page 106, specify the default primary and secondary quantities:

A In the Default Primary Quantity field, type the value for the primary quantity in cylinders.

B In the Default Secondary Quantity field, type the value for the secondary quantity in cylinders.

6 Specify the IBM Storage Management Subsystem (SMS) definitions for the optional SORTOUT data set classes:

A In the SMS Data Class field, type the name of the data class.

B In the SMS Storage Class field, type the name of the storage class.

C In the SMS Management Class field, type the name of the management class.

7 In the SORTPARM data set name field, type the name of the data set that provides parameters for SyncSort.

8 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.

Setting the JCL options for permanent data sets

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the individual data set options panels to specify information about the following permanent work data sets and image copy data sets:

- Sortout (WORKDDN or LOADDN)
- Sysut (WORKDDN)
- Copy (COPYDDN, RECOVERYDDN, RECOVERDDN, ICDDN, RECOVERYICDDN, OUTCOPYDDN, or FCOPYDD)
- Sysrec (UNLDDN, INDDN, or UNLOADDN)
- Archive (ARCHDDN)
- Cntl file (CNTLDDN)
- (CHANGE MANAGER only) Baseline recovery
- Discard (DISCARDDN)
- Error (ERRDDN)
- Map (MAPDDN)
- Report
- Punch (PUNCHDDN)
- Filter (FILTERDDN)

The permanent work data sets that contain data allow restarts. They are defined by using DISP=(NEW,CATLG) or DISP=SHR for restart or startover JCL. Examples include input (SYSUT), output (SORTOUT), discard (SYSDISC), map (SYSMAP), error (SYSERR), and punch (SYSPUNCH).

Other permanent data sets are used for restart and recover purposes. They use the same dispositions as the permanent work data sets. Examples include unload (SYSREC), copy (SYSCOPY), and baseline recovery (BLRP) (for CHANGE MANAGER only).

The ROWID and LOB SYSREC data sets are used only by the UNLOAD PLUS utility and the LOB DATA MOVER program in the Database Administration solution to unload and load data contained in a ROWID column and LOB columns.

**To set the JCL options for permanent data sets**

1. Use the following menu selections to display the panels for permanent work data sets:

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Individual Data Set Options</td>
</tr>
<tr>
<td>JCL Generation Individual Data Set Options Update</td>
<td>type of data set (SORTOUT, SYSUT, or COPY)</td>
</tr>
</tbody>
</table>

**Figure 39** shows the JCL Generation Data Set Options For Sortout Update panel.
Setting the JCL options for permanent data sets

**NOTE**

Not all of the options are available on all of the data set options panels.

If you choose to dynamically allocate your copy or unload data sets, any changes that you make to the copy (SYSCOPY), unload (SYSREC), and baseline recovery (BLRP) (for CHANGE MANAGER only) data set options in the JCL Generation override panels do not take effect.

---

**Figure 39  JCL Generation Data Set Options For Sortout Update panel**

```
----------- JCL GENERATION DATA SET OPTIONS FOR SORTOUT UPDATE -----------
COMMAND ===> 

Type data and press Enter. Press PF3 or END to return to the main panel.

Enter Data Set Prefix below:
  . . &PREFIX..&WKID..&STEPN

  Unit Name . . . . . . . . . SYSDA (SYSDA, TAPE, etc)
  Primary Space . . . . . . . 10 (Cylinders)
  Secondary Space . . . . . . 2 (Cylinders)
  Tape EXPDT. . . . . . . . .  (Blank or YYDDD or YYYY/DDD)
  Tape RETPD. . . . . . . . .  (Blank or 1 - 9999 days)
  SMS Data Class. . . . . . . (Blank or Data Class)
  SMS Storage Class . . . . . (Blank or Storage Class)
  SMS Management Class . . . (Blank or Management Class)

  Threshold Value . . . . . 0 (Cylinders, 0 means no Threshold)
  Alternate Unit Name . . . (SYSDA, TAPE, etc)
  Alternate SMS Data Class . (Blank or Data Class Name)
  Alternate SMS Storage Class (Blank or Storage Class Name)
  Alternate SMS Management Class (Blank or Management Class Name)
```

2 Specify the prefix for the data set.

To specify a GDG for the local and recovery image copy data sets, add the &GDG symbolic variable to the end of the data set prefix (Figure 40).
If you use the &GDG variable, JCL Generation resolves the data set name using the symbolic variable, and the name includes the GDG number (Figure 41).

Figure 41 Data set names resolved with the &GDG symbolic variable (Part 1 of 2)

```plaintext
//SYCLO001 DD DSN=RDACRJ.DEMOCJ.S9(+1),
   DCB=(SYS1.MODEL),
   DISP=(NEW,CATLG,CATLG),
   SPACE=(CYL,(10,2),RLSE),
   UNIT=SYSDA
//SYCLO002 DD DSN=RDACRJ.DEMOCJ.S3(+1),
   DCB=(SYS1.MODEL),
   DISP=(NEW,CATLG,CATLG),
   SPACE=(CYL,(10,2),RLSE),
   UNIT=SYSDA
//SYCLO003 DD DSN=RDACRJ.DEMOCJ.S2(+1),
   DCB=(SYS1.MODEL),
   DISP=(NEW,CATLG,CATLG),
   SPACE=(CYL,(10,2),RLSE),
   UNIT=SYSDA
//SYCLO004 DD DSN=RDACRJ.DEMOCJ.S11(+1),
   DCB=(SYS1.MODEL),
   DISP=(NEW,CATLG,CATLG),
   SPACE=(CYL,(10,2),RLSE).
```
In the **Unit Name** field, type the name of the unit.

Consider the following items when you specify the unit:

- If you use a tape unit, JCL Generation allocates the data sets before it opens them.
- For the Error data set (SYSERnnn) and Map data set (SYSMAP), specify SYSDA.
  
  If you specify TAPE and need to restart the IBM LOAD utility, you must uncatalog the existing data set. Then, you must change the DD statements to a disposition of (NEW,CATLG,CATLG).
- If you have large data sets and want to avoid extents or multiple data sets on DASD, specify a tape, virtual tape, or cartridge unit.
- As an alternative to specifying tape for the unit, you can specify a DASD unit that will span multiple data sets. Return to the JCL Generation Static Data Set Options panel (see page 105), and specify a value for **Max cylinders** and a value greater than 1 for **Max unit count**.
- BMC recommends that you specify **SYSDA** for the unit name for ROWID SYSREC unload data sets. (The ROWID SYSREC data set is used only by the UNLOAD PLUS utility and the LOB DATA MOVER program in the Database Administration solution.) If the ROWID SYSREC is set to SYSDA, the performance of the worklist commands for unloading the ROWID data set can be improved when the worklist is executed in parallel.

```bash
//         UNIT=SYSDA
/*--------------------------------------------------------------------
/*  SORT WORK DD STATEMENTS
/*--------------------------------------------------------------------
/SORTWK01 DD UNIT=SYSDA,
//   SPACE=(CYL,(10,2)),
//   DISP=(NEW,DELETE)
/SORTWK02 DD UNIT=SYSDA,
//   SPACE=(CYL,(10,2)),
//   DISP=(NEW,DELETE)
/SORTWK03 DD UNIT=SYSDA,
//   SPACE=(CYL,(10,2)),
//   DISP=(NEW,DELETE)
/SORTWK04 DD UNIT=SYSDA,
//   SPACE=(CYL,(10,2)),
//   DISP=(NEW,DELETE)
/SORTWK05 DD UNIT=SYSDA,
//   SPACE=(CYL,(10,2)),
//   DISP=(NEW,DELETE)
/*--------------------------------------------------------------------
```
4 If you typed N in the Data set sizing option field in step 2, specify the default primary and secondary quantities:

A In the Primary Space field, type the value for the primary quantity in cylinders.

B In the Secondary Space field, type the value for the secondary quantity in cylinders.

5 In the Tape EXPDT field, type the expiration date for a tape.

6 In the Tape RETPD field, type the retention date for a tape.

7 Specify the SMS definitions for the data set classes:

A In the SMS Data Class field, type the name of the data class.

B In the SMS Storage Class field, type the name of the storage class.

C In the SMS Management Class field, type the name of the management class.

8 In the Threshold Value field, type, in cylinders, the primary quantity for the data set.

If this value is exceeded, JCL Generation uses the alternate unit and the alternate SMS parameters. Zero indicates that a threshold is not specified for the unit. If you specify zero, JCL Generation does not use an alternate unit and the alternate SMS parameters.

9 In the Alternate Unit Name field, type the alternate name of the unit to be used if the threshold value specified in step 8 is exceeded.

NOTE

The alternate unit name must be a valid tape unit name. For more information, see “Setting the JCL options for tapes” on page 109.

10 Specify the SMS definitions for the alternate data set classes:

A In the Alternate SMS Data Class field, type the name of the data class.

B In the Alternate SMS Storage Class field, type the name of the storage class.

C In the Alternate SMS Management Class field, type the name of the management class.

11 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.
Setting the JCL generation data group options

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Generation Data Group Options panel to specify information about GDGs.

To set the JCL options for generation data groups

1 Use the following menu selections to display the JCL Generation Generation Data Group Options Update panel (Figure 42):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Generation Data Group Options (GDGs)</td>
</tr>
</tbody>
</table>

Figure 42 JCL Generation Generation Data Group Options Update panel

<table>
<thead>
<tr>
<th>COMMAND ===&gt;</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type data and press Enter.</td>
<td></td>
</tr>
<tr>
<td>Define GDG base at JCL generation? . N</td>
<td>(Y/N)</td>
</tr>
<tr>
<td>Specify NSCR on GDG definition? . N</td>
<td>(Y/N)</td>
</tr>
<tr>
<td>Number of primary copy GDG entries . 10</td>
<td>(1-255)</td>
</tr>
<tr>
<td>Number of recovery copy GDG entries . 10</td>
<td>(1-255)</td>
</tr>
<tr>
<td>Type GDG Model data set below:</td>
<td>. SYS1.MODEL</td>
</tr>
<tr>
<td>NOTE: GDGs only apply to copy data sets.</td>
<td></td>
</tr>
</tbody>
</table>

120 CATALOG MANAGER for DB2 User Guide
2 In the Define GDG base at JCL generation? field, type Y or N to specify whether JCL Generation creates the base of the GDG.

---

**NOTE**

JCL Generation cannot create the base of the GDG if you use IBM COPY to generate image copies for implicitly created objects.

---

3 In the Specify NSCR on GDG definition? field, type Y if the base of a GDG is defined in the IDCAMS DEFINE command as EMPTY (NSCR), or N if the base is defined as SCRATCH (SCR):

- SCR (the default) indicates to scratch (delete) the generation data set when it is uncataloged.
- NSCR indicates to uncatalog the generation data set when the maximum number of generation data sets to keep is reached.

4 In the Number of primary copy GDG entries field, type the maximum number of generation data sets to keep for primary copies.

5 In the Number of recovery copy GDG entries field, type the maximum number of generation data sets to keep for recovery copies.

6 At Type GDG Model data set below, type the name of the GDG model data set.

7 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.

---

**Setting the JCL debugging, display, and Execution options**

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Debugging, Display and Execution Options panel to specify information about how comments are handled in the JCL.
To set the JCL options for debugging, display, and Execution

1. Use the following menu selections to display the JCL Generation Debugging, Display And Execution Options Update panel (Figure 43):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Debugging, Display and Execution Options</td>
</tr>
</tbody>
</table>

Figure 43  JCL Generation Debugging, Display And Execution Options Update panel

```
--- JCL GENERATION DEBUGGING,DISPLAY AND EXECUTION OPTIONS UPDATE -------
COMMAND ==>

Type data and press Enter.

Include data set sizing comments in JCL . . . . . N (Y/N)
Include variable substitution comments in JCL . . . N (Y/N)
Suppress comments in JCL . . . . . . . . . . . . . N (Y/N)
NOTE: Do not set suppress comments to Y if you have specified either sizing or variable substitutions.
Specify an Alternate Program for IKJEFT01 . . . .
Post Step JCL INCLUDE member name . . . . . . .
Post Job JCL INCLUDE member name . . . . . . .
Include in AEXIN parameters:
SYNCDDELETE . . . N (Y/N)  BINDFAIL . . . N (Y/N)
HASHFAIL . . . N (Y/N)  HASHWARNRC . . . (NUMERIC)
REBINDFAIL . . . N (Y/N)  REBINDRC . . . (NUMERIC)
2MEGSQL . . . . . N (Y/N)  NOFAILNOIMAGECPY N (Y/N)
STOPWAIT . . . . 3 (NUMERIC)
STOPWAIT SECS . . 10 (NUMERIC)
```

2. Specify whether to include debugging comments in the generated JCL:

**NOTE**

BMC recommends that you include the comments if you suspect that the JCL was generated incorrectly and you need to send documentation to Customer Support. If you want to reduce the number of lines of JCL, do not include the comments.
In the Include data set sizing comments in JCL field, type Y or N to specify whether to include comments in the generated JCL that show statistics for determining data set sizes.

Comments are shown as \textit{dsso/cc}, where \textit{dsso} is the data set sizing option and \textit{cc} is a comment code. Table \ref{table:dsso} lists the comment codes that Execution generates in the JCL.

\begin{table}[h]
\centering
\caption{Comment codes for data set sizing}
\begin{tabular}{|l|c|p{0.3\textwidth}|}
\hline
Data set sizing options & Comment code & Description \\
\hline
B, C, or O & C & uses statistics from the DB2 catalog \\
 & H & uses the high relative-byte address (RBA) \\
 & M & uses multiple objects to size one data set (for example, SYSUTs) \\
 & N & indicates that the data set could not be sized because statistics could not be found \\
 & O & uses VSAM object sampling \\
 & S & uses the following formula to calculate the SORTWK size: \( \frac{(work space \times 2)}{number \ of \ SORTWK \ data \ sets} \) \\
 & W & warns that the sizing might be inaccurate \\
\hline
\end{tabular}
\end{table}

\textbf{Figure 44} shows example comments in the Execution JCL.

\textbf{Figure 44} Execution JCL with comments

```
/*--------------------------------------------------------------------
/*  ERRDDN OUTPUT DD STATEMENTS
/*--------------------------------------------------------------------
/* N = CAN'T SIZE DATA SET FOR DD SYSERO01, DEFAULTS USED BECAUSE
/*  NO BMCSTATS WERE FOUND FOR TB ACMX01.T.X01PS
/*SYSERO01 DD DSN=RDAMCG3.MG1217D.STEP1.SYSERO01,
 /*  DISP=(NEW,CATLG,CATLG),
 /*  SPACE=(CYL,(10,2),RLSE), ESTIMATE-B/N
 /*  UNIT=SYSDA

/*--------------------------------------------------------------------
/* SORT WORK DD STATEMENTS
/*--------------------------------------------------------------------
/* S = SORTWK SIZE IS (WORK SPACE * 2 / #SORTWKS)
/*SORTWK01 DD UNIT=SYSDA,
 /*  SPACE=(CYL,(1,1)), ESTIMATE-C/S
 /*  DISP=(NEW,DELETE)
```
Setting the JCL utility installation options module name options

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the BMC Utility Option Module Names panel to specify the name of the installation options module for the BMC utilities.

- In the **Include variable substitution comments in JCL** field, type **Y** or **N** to specify whether to include comments that show the SLIB variables and their assigned values.

  JCL Generation uses these variables to resolve the names of the data sets in the generated JCL.

3 In the **Suppress comments in JCL** field, type **Y** or **N** to specify whether to suppress all comments in the generated JCL.

**NOTE**

If you chose to include either the statistics comments or the variable comments in step 2, you cannot select to suppress all comments in the JCL.

4 In the **Specify an Alternate Program for IKJEFT01** field, type the name of a program to be used instead of IKJEFT01.

**NOTE**

The alternate program is only used for nonworklist JCL.

5 In the **Post Step JCL INCLUDE member name** field, type the name of a JCL member to be included after each step in the JCL.

6 In the **Post Job JCL INCLUDE member name** field, type the name of a JCL member to be included at the end of a job.

7 For each AEXIN keyword, type **Y** or **N** to specify whether to include the keyword in the AEXIN input stream.

8 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.
To set the JCL options for utility installation options module names

1. Use the following menu selections to display the JCL Generation BMC Utility Option Module Names Update panel (Figure 45):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER options processing</td>
<td></td>
</tr>
<tr>
<td>Y at Edit JCL Generation options</td>
<td></td>
</tr>
<tr>
<td>Y at Edit JCLgen options</td>
<td></td>
</tr>
<tr>
<td>Utility Options</td>
<td></td>
</tr>
<tr>
<td>BMC Utility Option Module Names</td>
<td></td>
</tr>
</tbody>
</table>

   **Figure 45  JCL Generation BMC Utility Option Module Names Update panel**

   ![Figure 45](image)

   Type module names and press Enter.
   Press PF3 or END to return to the main panel.

   - COPY PLUS . . . . . . Default ACP$OPTS
   - REORG PLUS. . . . . . Default ARU$OPTS
   - LOADPLUS. . . . . . . Default AMU$OPTS
   - UNLOAD PLUS . . . . Default ADU$OPTS
   - RECOVER PLUS. . . . Default AFR$OPTS
   - CHECK PLUS. . . . . Default ACK$OPTS
   - DASD MANAGER PLUS . Default blank

2. Specify the name of the installation options module for the BMC utilities.

   If you use any of the BMC utilities, the installation options module names will be listed in the AEXIN input stream in the JCL.

3. Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.
Setting the online reorg options

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Online Reorg Utility Options panel to specify the options for reorganizing table spaces. CATALOG MANAGER does not use these options.

To set the JCL options for online reorg options

1. Use the following menu selections to display the Online Reorg Utility Options panel (Figure 46):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Utility Options</td>
</tr>
<tr>
<td>Utility Options</td>
<td>Online Reorg Options</td>
</tr>
</tbody>
</table>

**Figure 46   Online Reorg Utility Options panel**

```
----------------------- ONLINE REORG UTILITY OPTIONS -----------------------
COMMAND ===>

BMCREORG XBMID:
REORG MAPPING TABLE:

NOTE: The mapping table full length cannot exceed 72 characters
```
2 In the BMCREORG XBMID field, specify the BMC EXTENDED BUFFER MANAGER (XBM) subsystem (SSID) that the BMC REORG PLUS utility accesses.

REORG PLUS uses XBM or its XBM SNAPSHOT UPGRADE FEATURE (SUF) technology to create a snapshot of the data sets to be reorganized.

The value of the SSID can be from 1 to 8 characters long.

3 In the REORG MAPPING TABLE field, specify the name of the mapping table that the IBM REORG utility uses to map the row IDs (RIDs) in the source table to the RIDs in the target table.

The name can be from 1 to 72 characters long.

4 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.

---

**Setting the non-worklist JCL options**

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the PROC and STEP Names panel to specify the name of the cataloged procedure (PROC) and the EXEC job step in the PROC for non-worklist JCL generated for utilities.

**To set the JCL options for non-worklist JCL**

1 Use the following menu selections to display the PROC and STEP Names panel (Figure 47):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Utility Options</td>
</tr>
<tr>
<td>Utility Options</td>
<td>Non worklist JCL PROC Options</td>
</tr>
</tbody>
</table>
2 In the **Use JCL Procedures (PROCS) for standard JCL?** field, type **Y** or **N** to specify whether to generate a PROC name instead of the EXEC PGM= statement in non-worklist JCL.

If you specify **Y**, you must provide a PROC name and a STEP name for the utilities for which you want to generate JCL.

---

**TIP**

You can specify the PDS that contains the PROCs in the **Jcllib** field on the JCL Generation Jobcard Options panel. To add JCLLIBs to the concatenation, modify the AJXJCLU SLIB. Alternatively, you can include the JCLLIB or PROCLIB in your jobcard JCL.

3 In the **Generate SET variables in JCL?** field, type **Y** or **N** to specify whether the AJXPSETV SLIB should generate SET statements in the JCL.

The AJXPSETV SLIB contains SET statements that assign SLIB variables to parameters. If you specify **Y**, JCL Generation generates SET statements for the parameters in the JCL.

---

**TIP**

You can add parameters or change the variable names in the AJXPSETV SLIB.
**NOTE**

SET statements in the PROC override those in the SLIB. To use SLIB parameters and variables in the SLIB, either remove those parameters from the PROC, or assign those parameters in the PROC to the variables in the SET statements in the SLIB. In addition, you can modify the statements in the AJXS$PROC SLIB for each utility.

4 In the **PROCNAME** field, specify the name of a PROC for the corresponding utility.

5 In the **PROCSTEP** field, specify the name of a STEP for the corresponding utility.

6 Press END to save your changes and return to the JCL Generation Update - Main Menu panel.

### Setting the LISTDEF and TEMPLATE data set options

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the IBM Utility Dynamic Data Set Options panel to specify the names of the data sets for TEMPLATE and LISTDEF definitions.

**To set the JCL options for LISTDEF and TEMPLATE data sets**

1 Use the following menu selections to display the JCL Generation IBM Utility Dynamic Data Set Options Update panel (Figure 48):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Listdef and Template Data Sets</td>
</tr>
</tbody>
</table>
Figure 48  JCL Generation IBM Utility Dynamic Data Set Options Update panel

2 At Enter External Template Data Set Name below, type the name of the data set that contains the TEMPLATE definitions that will be included in the JCL for the IBM utilities.

If you do not specify the name of a data set, JCL Generation allocates the data set to SYSTEMPL.

3 At Enter External Listdef Data Set Name below, type the name of the data set that contains the LISTDEF definitions that will be included in the JCL for the IBM utilities.

If you do not specify the name of a data set, JCL Generation allocates the data set to SYSLISTD.

4 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.

Setting user variables

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the User Defined Variables panel to specify character variables. Each variable has a corresponding symbolic variable that you can use in job cards or data set prefixes.
To set the JCL options for user-defined variables

1 Use the following menu selections to display the User Defined Variables panel (Figure 49):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Primary Menu panel</td>
<td></td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>User Defined Variable Values</td>
</tr>
</tbody>
</table>

Figure 49  User Defined Variables panel

```
---------------------- USER DEFINED VARIABLES UPDATE --
COMMAND ==>
Type data and press Enter.

Character Variables:
User variable 1 . . . . . . (Symbolic &UVR1)
User variable 2 . . . . . . (Symbolic &UVR2)
User variable 3 . . . . . . (Symbolic &UVR3)
User variable 4 . . . . . . (Symbolic &UVR4)
User variable 5 . . . . . . (Symbolic &UVR5)

NOTE: Symbolic variables cannot be input to these values
```

2 Specify the values for the variables.

The maximum length of a variable name is eight characters.

3 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.
Creating a user POF

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component. Use the Product Options File (POF) Functions panel to create a user POF or update the values in your ISPF profile. The panel displays the data set name of the initial POF. The panel also displays the value of the POFDATE parameter in the initial POF that was last used to update the ISPF profile.

1 Use the following menu selections to display the JCL Generation Product Options File (POF) Functions panel (Figure 50):

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Product Options File (POF) Functions</td>
</tr>
</tbody>
</table>

**Figure 50  JCL Generation Product Options File (POF) Functions panel**

<table>
<thead>
<tr>
<th>COMMAND =&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type data and press Enter. Press PF3 or END to return to the main panel.</td>
</tr>
</tbody>
</table>

- **Initial POF name:**
  - AUS.DOPSEC.CNTL(JX10QBDC)
  - Initial POF name Different from previous?  N
  - BROWSE Initial POF . . . . . . . . . . . N (Y/N)
  - MSGCLASS for POF Diagnostic Messages . X

- **POFDATE value used for initial POF Refresh Compare:** 2011/02/09 08:15:23

- **Type User POF Name below:**
  - AUS.DOPSEC.CNTL(JX10QBAC)
  - BROWSE, EDIT, VALIDATE User POF . . . . . . . . . . . N (B/E/V/N)
  - RESET Profile Variables from User POF that are marked with Refresh ".(R)" . . . . . . . . . . . N (Y/N)
  - RESET All Profile Variables from User POF . . . . . . . N (Y/N)
  - WRITE User POF data set from Profile Variables . . . . . . . N (Y/N)

2 In the **Type User POF Name below** field, replace the displayed name (the initial POF) with the name of the data set for a user POF.

The name can be either an existing sequential, 80-column data set or a member of a PDS.
3 In the **WRITE User POF data set from Profile Variables** field, type `Y` to write the ISPF variable values (located in the ISPF profile) to the user POF.

4 Press **Enter**.

---

### Updating a user POF

The product provides you two methods of updating a user POF: by using the options panels or by editing the file directly.

#### To update the user POF in the JCL Generation options panels

1 In the various options panels, specify your changes to the JCL Generation options.

2 Use the following menu selections to display the JCL Generation Product Options File (POF) Functions panel:

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Product Options File (POF) Functions</td>
</tr>
</tbody>
</table>

3 In the **Type User POF Name below** field, type the name of the data set for your user POF.

4 In the **WRITE User POF data set from Profile Variables** field, type `Y`.

5 Press **Enter**.

#### To update the user POF directly

1 From the JCL Generation Product Options File (POF) Functions panel, in the **Type User POF Name below** field, type the name of the data set for your user POF.
2 In the **BROWSE, EDIT, VALIDATE User POF field**, type **E**.

---

**NOTE**

You can edit the user POF or the initial POF by using the ISPF edit macro **AJXPODAT** from the `HLQ.DBCLIB` library. This library must be in your SYSPROC concatenation.

---

3 Edit and save the file.

4 In the **RESET All Profile Variables from User POF** field, type **Y** to update all of the ISPF variables in the ISPF profile with the variables in the user POF.

5 Press **Enter**.

---

**Using multiple POFs**

If you want to use different values for different applications, consider using more than one user POF. When you use multiple POFs, you can reset the values from a user POF that contains specifications for a particular application. Some sample scenarios follow:

**Scenario 1**

The Payroll department needs backup copies of their data sets on tape, but other departments do not need backup copies. You can create one user POF for the Payroll department and one for the other departments. Whenever you work with payroll objects, you can specify your payroll POF name to reset the profile variables from that POF.

**Scenario 2**

Your company is a service provider for several customers. By using a separate POF for each customer, you can accommodate each customer’s naming standards.

**Scenario 3**

You have different requirements for test and production data. For test data, you want to store the copies on DASD. For production data, you want to store the copies on tape. You can use separate POFs for test data and production data.
To create an additional user POF and reset the profile variables, perform the following steps:

1. Follow the steps in “Creating a user POF” on page 132 to create an additional POF.

2. From the JCL Generation Product Options File (POF) Functions panel, in the **Type User POF Name below** field, type the name of the data set for your customized POF.

3. In the **RESET All Profile Variables from User POF** field, type **Y** to update all of the ISPF variables in the ISPF profile with the variables in the user POF.

4. Press **Enter**.

Note that you would perform the same steps to specify use of a different user POF.

**Refreshing the initial POF**

Assume that your shop has revised its standards for naming data sets. To enforce the new standards, you need to refresh the initial POF so that users will get the updated values.

The JCL Generation component uses the value of the POFDATE keyword and the refresh attribute when determining whether to reset the ISPF profile variables to the updated POF values. The refresh attribute of a POF keyword value indicates that the ISPF profile variable should be reinitialized from the POF value if one of the following conditions exists:

- The value of the POFDATE keyword is greater than that saved in the ISPF profile.
- The name of a new initial POF is different from the name of the POF that is saved in the ISPF profile.

To refresh the initial POF, perform the following steps:

1. Edit the initial POF outside of the product.

2. Change the value of the POFDATE keyword to the current date.

3. Append the refresh attribute, *(R)* to the values that you want to update.

4. Save the POF.
After you save the changes, users will receive the updated ISPF variables the next time they invoke one of the products. However, *the users can still use their existing user POFs, which might not contain the updated values*. To ensure that the user POFs use the updated values, tell the users to reset all of their profile variables from the revised initial POF. If the users typically reset all of their variables from their user POFs, they must ensure that the updated values are included.

Alternatively, you can specify the name of a new initial POF in the POFDS installation option.

### Generating POF reports

Periodically, you might need to review POF keyword values, determine whether values are missing, or diagnose a problem. The following reports can assist you:

- **The POF Validation Report** lists a POF keyword, the action taken on the keyword, and the value of the keyword in the POF. For example, you can generate this report when you create a new user POF and change the values of several keywords. The report shows the changes, the number of values that were refreshed, and any errors that resulted.

- **The Variables Initialized with Default** report lists the keywords that are missing from the initial POF and the default ISPF variables that are used to populate the keywords. You can generate this report when you want to view the new keywords and their values for a release.

### To generate the reports

From the JCL Generation Product Options File (POF) Functions panel, in the **MSGCLASS for POF Diagnostic Messages** field, type the MSGCLASS for the SYSOUT field that is used to display messages. Consider using a SYSOUT class that is designated to go to the held queue so that you can view the output. Two SYSOUT files are allocated: AJXPOFER and AJXPOFVL.

**NOTE**

The default value for the **MSGCLASS for POF Diagnostic Messages** field is blank, which indicates that JCL Generation does not generate a report when you invoke the product.

When you invoke the product or reset the POF, the reports are listed on the output for your TSO session.
Reusing a POF in a subsequent installation

Assume that you customized the values in your POF, and now you are installing a new release of a product. To avoid having to customize the values again, you can specify that the Installation System use your existing POF to populate the values in the new initial POF. The new POF will contain your current values plus any new keywords (and their values) for the new release.

To reuse a POF, perform the following steps:

1. Run the Installation System.

2. From the Install System JCL Generation File Information panel, in the Use Existing POF to Populate the New Product Options File field, type Y (Figure 51).

3. Enter the names of the data set and member for the existing POF (Figure 52).

   For the member name, use the name of the POF that is used as the initial POF when you invoke the product.
Overriding POF values in SLIBs

One of the primary advantages of using POFs is that you can customize your JCL without having to modify your SLIBs. Nonetheless, you might need to modify your SLIBs from time to time. SLIB variables (or ISPF variables) are used in the SLIBs. Some of these SLIB variables correspond to the parameters in the POF. Note, however, that the names of the SLIB variables differ from the names of the POF keywords.

For example, if you specify the data set prefix for local primary copies (Figure 53), the name of the primary copy data set resolves to the following name without modifications to the SLIB:

\[
<SSIDname>.IC.T.ICPY.<databaseName>.<tableSpaceName>.<ddname>
\]

For data sets that are not dynamically allocated, JCL Generation appends the ddname to the prefix to create the name of the data set.
Now, assume that your site’s DBA decides that users should not have the ability to change the data set prefix from the options panels. The DBA can override the value in the SLIB. By specifying the value for the copy data set in the AJX#DSNS SLIB (Figure 54), the DBA can uphold your site’s naming standards.

Figure 54  Changing the SLIB variable for the copy data set in AJX#DSNS (Part 1 of 2)

```
)CM-------------------------------------------------------------------
)CM  SET SYSCOPY "PREFIX"
)CM-------------------------------------------------------------------
)SEL &AJXC1PRF ^= &Z
)SET SYSC1PR = &AJXC1PRF <- Resolved values from ISPF profile
)ENDSEL
)SEL &AJXC1PRF = &Z
)SEL &AJXSYCOP ^= &Z
)SET SYSC1PR = &AJXSYCOP
)ENDSEL
)SEL &AJXSYCOP = &Z
)SET SYSC1PR = &AJXHLQ..&AJXDB..&AJXTS
)ENDSEL
)ENDSL
)SEL &AJXC2PRF ^= &Z
)SET SYSC2PR = &AJXC2PRF
)ENDSEL
)SEL &AJXC2PRF = &Z
)SEL &AJXSYCOP ^= &Z
)SET SYSC2PR = &AJXSYCOP
)ENDSEL
)SEL &AJXSYCOP = &Z
```
After changing an SLIB variable, the DBA should use JCL Generation to test the changes. If the SLIB is coded correctly, the DBA must then recompile the SLIB. The DBA can use the SLIB compiler tool that is supplied with the Administrative products to compile the SLIB. For more information about testing the changes or using the SLIB compiler, see the product documentation.

### Adding steps to the JCL

You can include customized steps in the generated JCL (between product-generated steps or at the end of a job). For example, when you generate a sequence of utility steps, you might want to add a step to check the time or to send a message about the status of the job. You can do so by inserting JCL that is stored as a member of a partitioned data set. Use the JCLLIB statement to name the partitioned data set, and the INCLUDE statement to indicate where to include a member of the data set.

To add steps to the JCL, perform the following steps:

1. Use the following menu selections to display the JCL Generation Jobcard Options Update panel:

<table>
<thead>
<tr>
<th>From this menu</th>
<th>Select this item and press Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG MANAGER Primary Menu panel</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Options</td>
<td>Y at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td>Y at Edit JCLgen options</td>
</tr>
<tr>
<td>JCL Generation Update - Main Menu</td>
<td>Jobcard Options</td>
</tr>
</tbody>
</table>

```sql
SET SYSC2PR = &AJXHLQ..&AJXDB..&AJXTS
ENDSEL
ENDSEL
SET SYSC1PR = user.shop.standards.&AJXDB..&AJXTS  <- Override with SLIB variables and fixed values
```
2 From the JCL Generation Jobcard Options Update panel, in the **Jcllib** field, type the name of the partitioned data set and press END (Figure 55).

**Figure 55  JCL Generation Jobcard Options Update panel**

```plaintext
-------------------------------- JCL GENERATION JOBCARD OPTIONS UPDATE --------------------------------
COMMAND ===> 

Type data and press Enter.

Is a TSO submit exit used to generate jobcards? . . N (Y/N)
Enter Jobcards below:
//JOBCTRL JOB (&ZACCTNUM),`&PGMR',
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
// NOTIFY=&USERID
//*
//*
Jcllib . . . . . . SKH.INCLUDE.JCLLIB
Sysexec. . . . . .
Region size . . OM (See JCL Reference for valid options)
Time parameter . . (See JCL Reference for valid options)
System MLIB. . . . ISP.SISPMENU
Runtime HLQ. . . . AEX.QA0101
LLQ. . . . . . . . BMC
ULLQ . . . . . . . (Leave blank if using runtime enablement)
```

When you use the JCLLIB option, the JCL Generation component generates the following statement in the JCL (Figure 56):

**Figure 56  Including the JCL library**

```plaintext
//*-------------------------------------------------------
//* JCLLIB SPECIFIED
//*-------------------------------------------------------
// JCLLIB ORDER=SKH.INCLUDE.JCLLIB
```

3 From the JCL Generation Update - Main Menu, select **Debugging, Display and Execution Options** and press Enter.
4. From the JCL Generation Debugging, Display and Execution Options Update panel, specify the name of the member that contains the JCL that you want to run, and press END (Figure 57):

**Figure 57  JCL Generation Debugging, Display and Execution Options Update panel**

```plaintext
--------- JCL GENERATION DEBUGGING,DISPLAY AND EXECUTION OPTIONS UPDATE ---------
COMMAND ===> 

Type data and press Enter.

Include data set sizing comments in JCL . . . . . N (Y/N)
Include variable substitution comments in JCL . . . . . N (Y/N)
Suppress comments in JCL . . . . . . . . . . . . . . N (Y/N)
NOTE: Do not set suppress comments to Y if you have specified either sizing or variable substitutions.

Specify an Alternate Program for IKJEFT01 . . . .
Post Step JCL INCLUDE member name . . . . . . . . . STEPEND
Post Job JCL INCLUDE member name . . . . . . . . . . JOBEND

Include in AEXIN parameters:
SYNDELETE . . . . N (Y/N)  BINDFAIL . . . . N (Y/N)
HASHFAIL . . . . N (Y/N)  HASHWARNRC . . (NUMERIC)
REBINDFAIL . . . . N (Y/N)  REBINDRC . . . (NUMERIC)
2MEGSQL . . . . N (Y/N)  NOFAILNDIMAGECOPY N (Y/N)
STOPWAIT . . . . 3 (NUMERIC)
STOPWAIT SECS . . 10 (NUMERIC)
```

- If you want to run the JCL *between steps*, type the member name in the Post Step JCL INCLUDE member name field.

  The JCL Generation component generates the following statement in the JCL after each step (Figure 58).

**Figure 58  Including JCL between steps**

```plaintext
//*--------------------------------
//* END OF JOBSTEP
//*--------------------------------
//*--------------------------------
//* END OF STEP INCLUDE MEMBER
//*--------------------------------
// INCLUDE MEMBER=STEPEND
```

- If you want to run the JCL *at the end of the job*, type the member name in the Post Job JCL INCLUDE member name field.

  The JCL Generation component generates the following statement in the JCL at the end of the job (Figure 59).
Obtaining a list of TEMPLATEs or LISTDEFs in CATALOG MANAGER

You can use IBM TEMPLATE control statements to define the naming conventions and allocation of data sets. You can use IBM LISTDEF control statements to define lists of objects for utility processing. The POF provides options for specifying these control statements. For more information about using the TEMPLATE and LISTDEF control statements, see “TEMPLATE and LISTDEF control statements” on page 300.

Using the commands table

The commands that CATALOG MANAGER lists in the Commands List panel are defined in the primary commands table in the ACTCOMND member of the HLQ.DBCNTL library. This member contains all of the correct values for your installation. The contents vary among versions of CATALOG MANAGER. Figure 61 shows a portion of an ACTCOMND file.
Although you cannot modify ACTCOMND directly, you can create a user commands table. This table contains your modifications to existing commands and any new commands, and overrides the primary commands table. You specify your user commands table in the UCOMD installation option. When you start CATALOG MANAGER, the product merges the primary commands table with your user commands table.
Entries in the commands table use the following syntax:

Table 21  Syntax of commands table entry

<table>
<thead>
<tr>
<th>*commandName objectType (expandedName)</th>
<th>$ACTCMD commandName,$commandEquivalent,parameterName=value.</th>
</tr>
</thead>
</table>

Table 22 describes the variables in the commands table syntax.

Table 22  Commands table variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commandName</td>
<td>the name of the command that appears in the Commands List panel</td>
</tr>
<tr>
<td>objectType</td>
<td>the object’s two-, three-, or four-character abbreviation</td>
</tr>
<tr>
<td></td>
<td>The LIST command specifies an object type, such as LIST AL.</td>
</tr>
<tr>
<td>expandedName</td>
<td>the alternate or full name of the command (for example, ALIASES)</td>
</tr>
<tr>
<td>commandEquivalent</td>
<td>characters or numbers that represent a command</td>
</tr>
<tr>
<td></td>
<td>For example, #COL or 081 can represent the COLSTATS command. For a list</td>
</tr>
<tr>
<td></td>
<td>of valid characters and numbers, see the $ACTCEQU member in the HLQ:DBMAC</td>
</tr>
<tr>
<td></td>
<td>library.</td>
</tr>
<tr>
<td>parameterName</td>
<td>a parameter (for example, OBJECTS) that you use to define the CATALOG</td>
</tr>
<tr>
<td></td>
<td>MANAGER commands</td>
</tr>
<tr>
<td></td>
<td>For a list of valid parameters, see the $ACTCMD member in the HLQ:DBMAC</td>
</tr>
<tr>
<td></td>
<td>library.</td>
</tr>
<tr>
<td>value</td>
<td>a valid value for the parameter</td>
</tr>
</tbody>
</table>

A list of the parameters that can be used with $ACTCMD follows.

**NOTE**

The $ACTCMD member contains parameters that are not documented in this guide. These parameters are for CATALOG MANAGER internal use only.

&CLIST

This parameter indicates whether the command is a CLIST or REXX (YES/NO). If &CLIST=YES, the product issues the command as SELECT CMD(&LOAD). The default value is NO.
&CMD

This parameter specifies the command verb, should be the first parameter of the macro, and must be inserted in the table in alphabetical order. The maximum length is 16 bytes, and the command verb must be unique to the other command names. Embedded spaces are not allowed.

&DB2MAX

This parameter specifies the maximum (latest) version of DB2 for which the command is valid.

&DB2MIN

This parameter specifies the minimum (earliest) version of DB2 for which the command is valid.

&HELP

This parameter specifies which Help panel to display when the command is selected. The default is ACTHCMDS.

&LOAD

This parameter specifies the user load module or CLIST for the command. &LOAD is the entry point of the command processor.

&LOG

This parameter indicates whether any DDL command or DB2 command that initiates an action is logged in the Session Log. You must issue log requests from the program to use this feature. CATALOG MANAGER ships with ACTCOMND session logging enabled for all commands that support logging. If &LOG=NO, the log requests are ignored. The default value is NO.

&LSTO

This parameter indicates whether the command can be entered only against list items. If the value of &LSTO=YES, users must enter the command on the Command line. The default value is NO.

&NLIST

This parameter indicates whether to hide the command or to display it in the list that the COMMANDS command produces. &NLIST=YES hides the command from users. The default value is NO.
&NOSERVER

This parameter indicates whether the product can execute the command on the server that is in use. If &NOSERVER=YES, the product cannot use the command when you are connected to one DB2 subsystem and attached to another. The default value is NO.

&NUM

This parameter specifies the command ID number. User-written commands should be assigned command numbers greater than 1000.

--- NOTE ---

Do not change CATALOG MANAGER command numbers.

&OBJECTS

This parameter lists the two-character codes for the object types to which the command applies. Table 2 on page 38, Table 3 on page 39, and Table 4 on page 42 show the objects that the list can include.

--- NOTE ---

The use of object-type codes with the commands table is different from the use of object-type codes in CATALOG MANAGER functions. For example, SQ is valid for DBRMs in the commands table, but cannot be used to generate a list of DBRMs in CATALOG MANAGER.

Enclose the object list in parentheses and use a plus symbol (+) to separate each two-letter code.

&PARSE

This parameter indicates whether the command processor passes the command text without parsing it. If &PARSE=YES (the default), the text is scanned for an object type as the first token. If an object type is found, the field exccobjc is set to the type found.

&PLAN

This parameter specifies the plan to be opened and closed by CATALOG MANAGER when the command processor is called. If no plan is involved or the command processor opens and closes its own plans, you can set this parameter to NO (the default).
Creating a user commands table

You can create your user commands table by using the primary commands table, ACTCOMND, as a model. Copy the commands that you want to modify from the ACTCOMND member and paste them into a new member.

NOTE

You must include the following statements at the beginning of your user commands table:

COPY $ACTCEQU
$ACTCMD TYPE=DESECT

You can modify your user commands table by editing, disabling, or adding commands.

Specify the name of the user commands table in the UCOMD installation option. After creating the user commands table, you must compile and link the member. USRCOMND is the member in the HLQ.JCL and HLQ.UDBCNTL data sets that compiles and links the user commands table.

Editing commands in the commands table

You can edit a command in your user commands table to control how the command performs. For example, you can replace the format of one command with the format of another.

To replace the format of a command

1. Edit your user commands table in an ISPF edit session.

2. Replace the contents of one $ACTMD entry with the contents of another entry.
For example, assume that you do not have BMC utilities installed and want the STATUS command (Figure 62) to work like the DISPLAY UTILITY command (Figure 63). You can replace the format of the STATUS command with that of the DISPLAY UTILITY command.

**Figure 62  STATUS command**

<table>
<thead>
<tr>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ACTCMD STATUS, #STU, LOAD=ACTXXXXX, HELP=ACTHSTAU, CAT=YES, UTILCMD=YES</td>
</tr>
</tbody>
</table>

**Figure 63  DISPLAY UTILITY command**

<table>
<thead>
<tr>
<th>DISPLAY UTILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ACTCMD DISUTILITY, #DSU, LOAD=ACTXXXXX, HELP=ACTHKDUT, CAT=YES, UTILCMD=YES, OBJECTS=(DB+IC+IM+IX+SG+TB+TS)</td>
</tr>
</tbody>
</table>

If you replace everything after the pound sign (#) in the STATUS command with everything after the pound sign in $ACTCMD DISUTILITY, the new format of STATUS looks like the example shown in Figure 64.

**Figure 64  New STATUS command**

<table>
<thead>
<tr>
<th>DISPLAY STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ACTCMD STATUS, #DSU, LOAD=ACTXXXXX, HELP=ACTHKDUT, CAT=YES, UTILCMD=YES, OBJECTS=(DB+IC+IM+IX+SG+TB+TS)</td>
</tr>
</tbody>
</table>

3. Save the changes.

4. Run the USRCOMND job to compile and link your user commands table.

**To change the functionality of the BR and ED commands**

1. Edit your user commands table in an ISPF edit session.

2. Insert an asterisk (*) in column one for each row of the commands that you want to change.
Creating a user commands table

If you insert an asterisk for the BR and ED commands (that is, you comment the commands out) in the user commands table, you can enter the commands from a table or a view list to invoke the data browsing or data editing functions, respectively. However, when you do not comment out the BR and ED commands, you can use the commands from the following lists to invoke the IBM DB2 data editor (if it is installed):

- table
- view
- synonym
- alias

3 Save the changes.

4 Run the USRCOMND job to compile and link your user commands table.

Disabling commands

You can disable a command by using the #CMD_DISABLE command equivalent. You can also disable a command by commenting it out in the user commands table.

1 Edit your user commands table in an ISPF edit session.

2 Replace #commandEquivalent with #CMD_DISABLE.

3 Save the changes.

4 Run the USRCOMND job to compile and link your user commands table.

Adding commands to the commands table

Because CATALOG MANAGER uses a table for its commands, you can write and add your own commands to the product either as programs or as CLISTs.

To write the command as a program

1 Edit your user commands table in an ISPF edit session.

2 Create an entry for the user commands table. For information, see “Commands table syntax and parameters” on page 145.

3 Compile a load module for the command.

4 Bind a plan, if applicable.
To write the command as a CLIST

1. Edit your user commands table in an ISPF edit session.

2. Create an entry for the user commands table. For information, see “Commands table syntax and parameters” on page 145.

3. Write the CLIST. For more information, see “Writing user commands as CLISTs” on page 152.

4. Run the USRCOMND job to compile and link the user commands table.

Retaining a user commands table when upgrading CATALOG MANAGER

During product installation, you can copy an existing user commands module from an existing library to a new library. This allows you to preserve a customized commands table from a previous release when upgrading the product.

To retain a user commands table from a previous CATALOG MANAGER release and optionally make changes

1. Perform one of the following tasks:

   ■ To copy a commands table that you used for CATALOG MANAGER version 9.1 or earlier, copy `HLQ.LOAD(userCommandsTable)` from that release to the latest release of the `HLQ.UDBLINK` library.

   ■ To copy a commands table that you used for CATALOG MANAGER version 9.1.01 or later, copy `HLQ.UDBLINK(userCommandsTable)` from that release to the latest release of the `HLQ.UDBLINK` library.

2. (optional) Perform one of the following tasks:

   ■ To copy a commands table that you used for CATALOG MANAGER version 9.1 or earlier, copy `HLQ.CNTL(userCommandsTable)` from that release to the latest release of the `HLQ.UDBCNTL` library.

   ■ To copy a commands table that you used for CATALOG MANAGER version 9.1.01 or later, copy `HLQ.UDBCNTL(userCommandsTable)` from that release to the latest release of the `HLQ.UDBCNTL` library.

3. (optional) Make any additional changes to the user commands table source.

4. Run the USRCOMND job to compile and link the ACTCOMND member.
Writing user commands as CLISTs

You can write CLISTs for user commands. Parameters are passed in CLIST variables (ISPF sharepool variables) with the same names as those of the programming parameter list.

**NOTE**

When you use a CLIST, you must set &WFEK=NO. CLISTs cannot use the WFEK option. In addition, CLISTs cannot write to the DDL Audit Log, Session Log, or Drop Recovery Log.

---

Development aids for user commands

Table 23 lists the members that are shipped with the product to aid in the development of your own CATALOG MANAGER commands.

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTCOMND</td>
<td>original source code for the commands table</td>
</tr>
<tr>
<td>ACTUSER</td>
<td>sample user command program in ASM</td>
</tr>
<tr>
<td>$ACTEXC</td>
<td>macro DSECT for user program variables</td>
</tr>
<tr>
<td>$ACTULOG</td>
<td>macro for user logging</td>
</tr>
<tr>
<td>$ACTLISTC</td>
<td>sample CLIST that displays IDCAMS LISTCAT output for a given data set</td>
</tr>
<tr>
<td>$ACTVARS</td>
<td>sample CLIST that shows how the variables are set when you execute a CLIST from a list</td>
</tr>
<tr>
<td></td>
<td>To view $ACTVARS, type <strong>CLIST ACTVARS</strong> in the <strong>Cmd</strong> column of a list, and then press <strong>Enter</strong>. The command displays a panel that shows the value of all variables.</td>
</tr>
<tr>
<td>USRCOMND</td>
<td>JCL to assemble and link the ACTCOMND source code for the commands table</td>
</tr>
</tbody>
</table>
Command program parameters

A user command program is invoked using standard operating system calling conventions:

- save area in register 13
- address in register 15
- return in register 14

Register 1 points to the user program parameter list. The macro $ACTEXC provides the DSECT of the parameter list. This macro is in the HLQ.DBMAC library.

Table 24 lists the key parameters.

Table 24  Key parameters for a user command program

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCRC</td>
<td>ISPF END function to execute on return</td>
</tr>
<tr>
<td>EXCTSOID</td>
<td>TSO ID that is running at the time</td>
</tr>
<tr>
<td>EXCSSID</td>
<td>attached DB2 subsystem ID</td>
</tr>
<tr>
<td>EXCPLAN</td>
<td>DB2 plan name</td>
</tr>
<tr>
<td>EXCESID</td>
<td>CATALOG MANAGER session ID</td>
</tr>
<tr>
<td>EXCCOMND</td>
<td>name of the command that is being executed</td>
</tr>
<tr>
<td>EXCCSRC</td>
<td>where the command is entered:</td>
</tr>
<tr>
<td></td>
<td>- S - list line</td>
</tr>
<tr>
<td></td>
<td>- C - Command line</td>
</tr>
<tr>
<td>EXCCOBJL</td>
<td>object list type where the command is valid</td>
</tr>
<tr>
<td>EXCCOBJC</td>
<td>object type found in the command text if PARSE=YES</td>
</tr>
<tr>
<td>EXCCLOG</td>
<td>command table log option</td>
</tr>
<tr>
<td>EXCCNUM</td>
<td>number of objects selected</td>
</tr>
</tbody>
</table>

Some fields are provided for interaction with the CATALOG MANAGER log routines, but you should not modify those fields.

For logging, use the $ACTULOG macro with the parameters shown in Table 25.
Passing object type and name

The object type is passed in fields EXCCOBJL and EXCCOBJC. The first field shows the object type of the list on which the command was entered. The second field is the same as the first, unless you entered an object type keyword as the first parameter of the command and the commands table specifies &PARSE=YES.

The name of the object is passed in various name fields depending on the type of the object selected. Due to the wide variety of name structures, these name fields are mapped in the DSECT with many overlays. The object type in EXCCOBJL determines the mapping. If you enter the command on the Command line, these fields are blank and you must parse the entered command for the object name.

If you enter the command, object type, and a name in the Cmd column of a list line, the name fields are populated with the name of the object on the line originally. If EXCCOBJL and EXCCOBJC are different, you must parse the entered command text for the object name even though the name fields are populated.

Table 25 Parameters for use with the $ACTULOG macro

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| pointer   | points to a log text field  
The field should be in the format LLtext where LL is the length of the field including itself. The text can be whatever you want put in the log. |
| OPT       | specifies which logs to write to. The default value is S. The options are as follows:  
- A - Audit Log  
- S - Session Log  
- B - both the Audit Log and the Session Log |
| FUNC      | address of a field containing the 16-byte function to log  
It does not need to match the command name. If this parameter is not used, blanks are logged. |
| RC        | identifies the register containing the return code to be logged  
If this parameter is not used, X’00’ is logged as the return code. |
| TYPE      | address of a 10-byte value to be logged as the object type  
If this parameter is not used, blanks are logged. |
| QUAL      | address of an 8-byte value to be logged as the object name qualifier  
If this parameter is not used, blanks are logged. |
| NAME      | address of an 18-byte value to be logged as the object name  
If this parameter is not used, blanks are logged. |
If you invoke the CLIST command on an object type that is not supported, CATALOG MANAGER places an appropriate message in the ISPF EXCOFNM0 variable. If you invoke the CLIST command on an invalid CLIST name, CATALOG MANAGER issues a message and continues to execute.

Table 26 lists the object types and their corresponding name fields.

<table>
<thead>
<tr>
<th>Object</th>
<th>Type</th>
<th>EXCOFNM0</th>
<th>EXCOFNM1</th>
<th>EXCOFNM2</th>
<th>EXCOFNM3</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>alias</td>
<td>AL</td>
<td>NA</td>
<td>creator</td>
<td>name</td>
<td>NA</td>
<td>EXCOFNMDB (DB name)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNTS (TS name)</td>
</tr>
<tr>
<td>column</td>
<td>CO</td>
<td>column</td>
<td>TB creator</td>
<td>TB name</td>
<td>column</td>
<td>NA</td>
</tr>
<tr>
<td>data set</td>
<td>DS</td>
<td>DS name</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>database</td>
<td>DB</td>
<td>NA</td>
<td>DB name</td>
<td>NA</td>
<td>NA</td>
<td>EXCOFNMDB (DB name)</td>
</tr>
<tr>
<td>DBRM</td>
<td>DM</td>
<td>NA</td>
<td>DM plan</td>
<td>DM plan</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>image copy</td>
<td>IC</td>
<td>NA</td>
<td>DB name</td>
<td>TS name</td>
<td>date</td>
<td>EXCOFNMDB (DB name)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNTS (TS name)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNM4 (time)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNMID (date)</td>
</tr>
<tr>
<td>index</td>
<td>IX</td>
<td>NA</td>
<td>creator</td>
<td>name</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>IX partition</td>
<td>IP</td>
<td>NA</td>
<td>IX creator</td>
<td>IX name</td>
<td>partition</td>
<td>NA</td>
</tr>
<tr>
<td>key column</td>
<td>KC</td>
<td>column</td>
<td>IX creator</td>
<td>IX name</td>
<td>column</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>sequence</td>
</tr>
<tr>
<td>materialized</td>
<td>MQ</td>
<td>NA</td>
<td>creator</td>
<td>name</td>
<td>NA</td>
<td>EXCOFNMDB (DB name)</td>
</tr>
<tr>
<td>query table</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNTS (TS name)</td>
</tr>
<tr>
<td>package</td>
<td>PG</td>
<td>NA</td>
<td>name</td>
<td>collection ID</td>
<td>NA</td>
<td>EXCOFNM4 (version)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNM5 (contoken)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNM6 (location)</td>
</tr>
<tr>
<td>plan</td>
<td>PL</td>
<td>NA</td>
<td>name</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>sequences</td>
<td>SE</td>
<td>NA</td>
<td>schema</td>
<td>name</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>storage group</td>
<td>SG</td>
<td>NA</td>
<td>name</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>synonym</td>
<td>SY</td>
<td>NA</td>
<td>creator</td>
<td>name</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>table</td>
<td>TB</td>
<td>NA</td>
<td>creator</td>
<td>name</td>
<td>NA</td>
<td>EXCOFNMDB (DB name)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNTS (TS name)</td>
</tr>
<tr>
<td>table space</td>
<td>TS</td>
<td>NA</td>
<td>DB name</td>
<td>name</td>
<td>NA</td>
<td>EXCOFNMDB (DB name)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNTS (name)</td>
</tr>
<tr>
<td>trigger</td>
<td>TR</td>
<td>NA</td>
<td>schema</td>
<td>name</td>
<td>NA</td>
<td>EXCOFNM4 (TB owner)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNM5 (TB name)</td>
</tr>
<tr>
<td>TS partition</td>
<td>TP</td>
<td>NA</td>
<td>DB name</td>
<td>TS name</td>
<td>partition</td>
<td>EXCOFNMDB (DB name)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNTS (TS name)</td>
</tr>
<tr>
<td>view</td>
<td>VW</td>
<td>NA</td>
<td>creator</td>
<td>name</td>
<td>NA</td>
<td>EXCOFNMDB (DB name)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EXCOFNTS (TS name)</td>
</tr>
<tr>
<td>volume</td>
<td>VL</td>
<td>NA</td>
<td>volume ID</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>user</td>
<td>UA</td>
<td>NA</td>
<td>grantee</td>
<td>grantor</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>US</td>
<td>NA</td>
<td>grantee</td>
<td>grantor</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
Table 27 describes the ISPF variables that you can access from user-written commands or CLISTs.

**Table 27  ISPF variables for user commands or CLISTs**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;VCAT</td>
<td>8</td>
<td>high-level qualifier of the VSAM catalog (VCAT)</td>
</tr>
<tr>
<td>&amp;SSID</td>
<td>4</td>
<td>the DB2 subsystem Identifier</td>
</tr>
<tr>
<td>&amp;ACTSRVR</td>
<td>16</td>
<td>the active server</td>
</tr>
<tr>
<td>&amp;ACTCOLID</td>
<td>18</td>
<td>the active collection identifier</td>
</tr>
</tbody>
</table>

**NOTE**  
To prevent accidental updates to the data in Table 28, set the status of the table spaces to Read Only.

The tables that are listed in Table 28 are created by CATALOG MANAGER. You can refer to the data in these tables for information on your daily operations.

**Table 28  CATALOG MANAGER tables (Part 1 of 2)**

<table>
<thead>
<tr>
<th>Table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCACT_vr.VEVR_ATTR</td>
<td>contains information about attributes</td>
</tr>
<tr>
<td>BMCACT_vr.VEVR_ATTR_VAL</td>
<td>contains the valid values for each attribute</td>
</tr>
<tr>
<td>BMCACT_vr.VEVR_AUDIT_LOG</td>
<td>contains an entry for each DDL statement that is executed by CATALOG MANAGER to modify the DB2 catalog</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Using the DDL Audit Log” on page 355.</td>
</tr>
<tr>
<td>BMCACT_vr.VEVR_CRS_VAL</td>
<td>contains cross-validation between attributes</td>
</tr>
<tr>
<td>BMCACT_vr.VEVR_DLG</td>
<td>contains dialog-specific information</td>
</tr>
<tr>
<td>BMCACT_vr.VEVR_DLG_ATTR</td>
<td>contains the attributes to be displayed for each dialog</td>
</tr>
</tbody>
</table>
Where to go from here

With CATALOG MANAGER set up to help you work as productively as possible, you are ready to learn how CATALOG MANAGER interacts with the DB2 subsystems that you use. Chapter 4, “Accessing other DB2 subsystems,” explains how to change the default SSID attachment in order to work with the catalog on another SSID.

### Table 28  CATALOG MANAGER tables (Part 2 of 2)

<table>
<thead>
<tr>
<th>Table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCACTvr.Vr_EDITOR_USERS</td>
<td>contains one row for each data editing session in progress. A row is inserted when data is fetched and deleted when the editing session ends. All rows that reference a particular user ID are deleted when that user ID begins another editing session. If an editing session ends abnormally, the applicable row might not be deleted from the table. Users with SYSADM authorization can delete all rows from the table by using an option on the Edit and Browse Options panel. For more information, see “Clear Editor Users Table” on page 229.</td>
</tr>
<tr>
<td>BMCACTvr.Vr_FCRS</td>
<td>contains the Filter Combo Result Table</td>
</tr>
<tr>
<td>BMCACTvr.Vr_GMAP</td>
<td>contains the grid mapping table</td>
</tr>
<tr>
<td>BMCACTvr.Vr_MSG</td>
<td>contains the informational messages</td>
</tr>
<tr>
<td>BMCACTvr.Vr_RECOVERY_LOG</td>
<td>contains an entry for each object that is dropped and each privilege that is revoked when the DROP statement is executed within CATALOG MANAGER and the Drop Recovery option is set to Y. For more information, see “Using the Drop Recovery Log” on page 355.</td>
</tr>
<tr>
<td>BMCACTvr.Vr_SEARCH_VARS</td>
<td>contains the values for variables that were entered in SEARCH statements and saved. For more information, see “Using host variables in a search” on page 190.</td>
</tr>
<tr>
<td>BMCACTvr.Vr_SESSION_LOG</td>
<td>contains an entry for each CATALOG MANAGER action for which logging was requested. For more information, see “Using the Session Log” on page 351.</td>
</tr>
<tr>
<td>BMCACTvr.Vr_SQL_TABLE</td>
<td>contains SQL statements that have been saved. For more information, see “Working with the SQL_Table” on page 213.</td>
</tr>
<tr>
<td>BMCACTvr.Vr_VIEW</td>
<td>contains the initial view for each utility that is supported</td>
</tr>
</tbody>
</table>
Accessing other DB2 subsystems

This chapter contains the following topics:

Overview ................................................................. 159
Using the DB2 Attach feature ........................................ 160
  Attaching CATALOG MANAGER to a specified SSID ........... 161
  Attaching to an SSID or server by using the connection selection list .... 162
Switching catalog access .............................................. 163
Restoring the default attachment .................................. 164
Using the DB2 Connect feature ...................................... 164
  Connecting to a specified SSID .................................... 166
Obtaining data set information from a remote SSID ............. 168
Connecting to a remote SSID from a location list ................ 169
Connecting to a remote SSID and issuing DB2 commands to display, start, or stop objects ..................................................... 169
Using saved connections .............................................. 170
Viewing the connections table ...................................... 173
Using DB2-identifiers with the CONNECT command ............ 174
Identifying attachments or connections with unique values .... 175
Displaying the current server ID .................................... 175
Troubleshooting a DB2 Attach or DB2 Connect failure ........... 176
Where to go from here ................................................ 176

Overview

A CATALOG MANAGER session is always attached to a single DB2 subsystem (SSID) on the same z/OS system. You might connect the SSID to another single SSID in order to query its catalog. CATALOG MANAGER provides the following features for you to substitute either or both of these SSIDs without closing your CATALOG MANAGER session:

- Use the DB2 Attach feature to modify the attachment between CATALOG MANAGER and an SSID.
Use the DB2 Connect feature to establish or modify a connection between the attached SSID and another SSID.

**Using the DB2 Attach feature**

When you start CATALOG MANAGER, the session is attached to a single DB2 subsystem as defined in the invoking BMCDB2 CLIST. The DB2 Attach feature enables you to end the attachment between CATALOG MANAGER and the current SSID, and then establish an attachment to a different SSID on the same z/OS system. You are still able to use all CATALOG MANAGER features.

DB2 Attach has the following requirements:

- CATALOG MANAGER must be installed on the target z/OS system at the same maintenance level as the product instance from which you are connecting.
- The target SSID must be on the same z/OS system as CATALOG MANAGER.
- The DB2 Call Attach Facility (CAF) must be installed.

When you attach to a different subsystem, the installation options and profile that were invoked on your original CATALOG MANAGER session are still in effect. If you make changes to your options while attached to another DB2 subsystem, the changes apply to your original session as well. Figure 65 shows a current attach and a possible attach of a CATALOG MANAGER session to another DB2 subsystem.

**Figure 65  Attaching to another DB2 subsystem**
Attaching CATALOG MANAGER to a specified SSID

The following procedure describes how to attach a CATALOG MANAGER session to a different SSID on the same z/OS system.

1. From the Primary Menu panel, enter CONNECT (CON) on the Command line.

The Change Access panel is displayed. (Figure 66).

**Figure 66  Change Access panel showing DB2 attach specifications**

<table>
<thead>
<tr>
<th>DB2V</th>
<th>Current Value</th>
<th>Change to</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSID</td>
<td>10 DEFF</td>
<td>ACT101DM</td>
<td>Call Attach</td>
</tr>
<tr>
<td>Plan</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server</td>
<td></td>
<td>Current Server</td>
<td></td>
</tr>
<tr>
<td>SQLid</td>
<td>ROACRJ2</td>
<td>SQLid on Server</td>
<td></td>
</tr>
<tr>
<td>Server SSID</td>
<td></td>
<td>SSID of Server</td>
<td></td>
</tr>
<tr>
<td>Collection</td>
<td></td>
<td>Direct/Indirect</td>
<td></td>
</tr>
</tbody>
</table>

2. In the Change to column, type the values that identify the attachment that you want to implement.

   **A** In the SSID field, type the SSID of the DB2 to which you want to attach.

   **B  (optional)** In the Plan field, type the plan name.

   The plan must be bound on the SSID to which you want to attach. The default plan name is specified in the MPLAN installation option.

3. Press Enter.

The Change Access panel is refreshed. The last value that was changed is displayed in the short message area of the panel.
Attaching to an SSID or server by using the connection selection list

You can also select values for the **SSID**, **Plan**, **Server**, and **Collection** fields from the connection selection list that is created during the installation of CATALOG MANAGER.

1. From the Primary Menu panel, enter `CONNECT (CON)` on the Command line.

   The Change Access panel is displayed (Figure 66).

2. In the **SSID** field, type a question mark (?).

3. In the **Plan** field, type a question mark (?).

4. Press Enter.

   The Connection Selection List panel is displayed with available SSIDs. The content of the connection selection list is created in the BMCDB2 CLIST when CATALOG MANAGER is installed.

5. In the **Sel** column, type **S** beside the SSID to which you want to attach.

6. Press Enter.

   The Connection Selection List panel is displayed. The short message area shows that you are now attached to the selected SSID. The panel lists the plan names that are available for the SSID that you have selected.

7. In the **Sel** column, type **S** beside the plan that you want to select.

---

**NOTE**

If the SSID that you select is incompatible or otherwise unavailable, a warning message is displayed in the short message area of the Change Access panel.
8 Press Enter.

The Change Access panel is displayed, showing the newly attached SSID and plan name in the Current Value column.

9 Press END to display the Primary Menu panel.

**Switching catalog access**

Catalog indirection, an optional method of implementing CATALOG MANAGER and other Administrative products from BMC, reduces contention for the DB2 catalog and improves performance by enabling users to access the catalog indirectly through copies or views. Catalog indirection can also support high security by restricting access to specified columns within the catalog tables.

The system administrator can set up CATALOG MANAGER to access either a real (direct) or indirect catalog at startup; users can also switch between the real and indirect catalogs. This procedure describes how to switch access between a real catalog and an indirect catalog.

1 From the Primary Menu panel, enter CONNECT (CON) on the Command line.

The Change Access panel is displayed.

2 In the Collection field, type a question mark (?).

3 Press Enter.

The Connection Selection List panel is displayed.

4 To select a collection, type S in the Sel column beside the collection name.

5 Press Enter.

CATALOG MANAGER now accesses the real or indirect catalog to which the selected collection points.

The panel ID area of any CATALOG MANAGER panel displays the SSID to which you are attached. If the SSID is followed by -R, the real catalog is being used; -I indicates that an indirect catalog is being used.
**Restoring the default attachment**

You can restore the attachment that was in effect at the start of the current CATALOG MANAGER session by using either of the following commands:

- From the Primary Menu panel, enter **CONNECT RESET** on the **Command** line.
- On the Change Access panel, enter **RESET** on the **Command** line.

After you restore the attachment, refresh the product options file (POF). For more information, see “Refresh the initial POF” on page 135.

**Using the DB2 Connect feature**

The DB2 Connect feature enables you to connect your attached SSID to another SSID through the DB2 Distributed Data Facility (DDF) in order to execute SQL. DB2 Connect has the following requirements:

- CATALOG MANAGER must be installed on the target z/OS system at the same maintenance level as the product instance from which you are connecting.
- The IBM SYSPROC.DSNWZP stored procedure must be installed. For more information, see the IBM DB2 for z/OS Installation Guide.
- The DB2 Distributed Data Facility (DDF) must be installed.
- The target subsystem can be on the same system or on a different z/OS system.
- The target subsystem can be at any level of DB2.

CATALOG MANAGER provides the following features:

- If catalog indirection was set up when CATALOG MANAGER was installed, you can connect to other catalog copies or views.
- If you install the IBM SYSPROC.ADMIN_DS_LIST stored procedure, you can obtain data set information on the remote SSID. For more information, see “Obtaining data set information from a remote SSID” on page 168.
- If you install the IBM SYSPROC.ADMIN_COMMAND_DB2 stored procedure, you can issue the DB2 DISPLAY, START, and STOP commands on the remote SSID. For more information, see “Connecting to a remote SSID and issuing DB2 commands to display, start, or stop objects” on page 169.

If you issue the DB2 commands on a *local* SSID, the product calls the IBM Instrumentation Facility Interface (IFI).
NOTE

When you are connected to a remote SSID, CATALOG MANAGER does not allow the following commands:

- SPACE and STATS commands that invoke DASD MANAGER PLUS
- DSN commands such as BIND, REBIND, DCLGEN, or FREE

Figure 67 shows how CATALOG MANAGER is attached to DB2A and connected to DB2B. CATALOG MANAGER could be connected to DB2C through the session attached on DB2A.

Figure 67  Attaching and connecting to other DB2 subsystems

```
  z/OS 1
  CATALOG MANAGER  Attach

  SSID.......DB2A
  Plan.......ACT930DM
  Server........DB2B_SERVER
  or
  DB2C_SERVER
  Collection. ACT930_D_MAIN

  z/OS 2
  At any level of DB2

  SSID.......DB2C
  Plan.......ACT930_DM
  Server........DB2C_SERVER
  Collection. ACT930_D_MAIN
```

Type the server name. If target subsystem is in another z/OS system, specify collection also.
Connecting to a specified SSID

You can connect the attached SSID to another SSID by using one of the following methods:

- specifying access values on a panel
- specifying the CONNECT command and parameters

**To specify the access values on a panel**

1. From the Primary Menu panel, enter CONNECT (CON) on the Command line.

   The Change Access panel is displayed.

2. In the **Server** field, type the server name or location name.

   A **Server** connection enables you to perform actions that are accomplished by using SQL.

3. In the **Collection** field, type the collection name for the catalog, catalog view, or catalog copy that you want to use.

   **NOTE**
   You can enter a question mark (?) in the **Server** and **Collection** fields. CATALOG MANAGER displays a Connection Selection List panel from which you can select a server and collection name.

4. *(optional)* In the **Server SSID** field, type the SSID of the server to which you want to connect.

   **NOTE**
   Enter the correct **Server SSID** to enable generation of batch jobs such as utilities, BIND, REBIND, and FREE. Before you submit any jobs for execution, ensure that the JCL has the correct STEPLIB and routing information.

5. Press **Enter**.

   The Change Access panel is displayed *(Figure 68)*. The collection ID that CATALOG MANAGER is using on the connected SSID is displayed in the **Current Value** column and in the short message area of the panel.
Figure 68  Change Access panel after new connection

| DEFF-S ------------------------ Change Access --------------- DBDC IS SERVER |
|-------------------------------|-----------------------------|
| Command ====>                |
| Display quick connections list  | . . . . .  N (Y/N)         |
| Display product plans and connections table  | N (Y/N) |
| The left column shows the current values of your CATALOG MANAGER attach and connection. Enter in the right column the changes you want to make. If a "Change to" field is blank or the same as the "Current Value" field no action will be taken. If you are not sure of the correct values to enter, put a question mark (?) in the field for a selection list. |
| The RESET command will attach you to your original SSID and open the original plan, set to your original collection and reset the server to blanks. |
| DB2V  | Current Value | Change to | Changes |
| SSID  | 10   | DEFF   | Call Attach |
| Plan  | ACT101DM |
| Server  | 91   | DBDC   | Current Server |
| SQLid  | RDACRJ | SQLid on Server |
| Server SSID  | DBDC   | SSID of Server |
| Collection  | ACT_QA1010 | Direct/Indirect |

6  Press END to display the Primary Menu panel.

The current connection is noted at the bottom of the panel.

To specify the CONNECT command and parameters

From the Primary Menu panel, enter CONNECT (CON) parameters on the Command line. Table 29 defines the parameters.

Table 29  CONNECT command parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>identifies the remote DB2 SSID</td>
</tr>
<tr>
<td>collection</td>
<td><em>(optional)</em> names the collection and package set for the remote DB2 SSID</td>
</tr>
<tr>
<td>SQLID</td>
<td><em>(optional)</em> specifies the current SQLID for the remote DB2 SSID</td>
</tr>
</tbody>
</table>
Obtaining data set information from a remote SSID

For the following examples, assume that the value of location is DEEG.

**EXAMPLE**

*Using the default collection and SQLID:*

```
CONNECT DEEG
```

*Using the default collection for SQLID RDA CRJ:*

```
CONNECT DEEG ? RDA CRJ
```

*Using the ACT101 collection for SQLID RDA CRJ:*

```
CONNECT DEEG ACT101 RDA CRJ
```

*Using the ACT101 collection for the default SQLID:*

```
CONNECT DEEG ACT101
```

Obtaining data set information from a remote SSID

When you are connected to a remote SSID with the CONNECT command, you can obtain data set information on the remote SSID by using the IBM SYSPROC.ADMIN_DS_LIST stored procedure.

**To install the DB2 stored procedure**

1. Create a JCL startup procedure for the IBM z/OS Workload Manager (WLM) environment.

2. Create the SYSPROC.ADMIN_DS_LIST stored procedure in the DB2 catalog, and specify the WLM environment.

3. Activate the WLM environment.

For more information, see the IBM *DB2 for z/OS Installation Guide*.

**To obtain data set information**

1. On your local DB2 subsystem, create an object list.

   For more information, see Chapter 2, “Getting started with CATALOG MANAGER.”

2. From the object list, type CONNECT on the Command line.

   The Change Access panel is displayed.

3. In the Server field, type the name of the remote server.
Connecting to a remote SSID from a location list

To connect to a remote DB2 SSID from a location (LO) list, perform the following steps:

1. To list locations, enter **LO** on the **Command** line.

2. In the **Cmd** field, enter **CONNECT** next to the remote location to which you want to connect.

**TIP**
You can also connect to a remote SSID by issuing the CONNECT BATCH command on a location list or by editing the SYSIN input stream and specifying the CONNECT command. For more information, see “Using the BATCH command for a DB2 object list or a mixed list” on page 207.

Connecting to a remote SSID and issuing DB2 commands to display, start, or stop objects

When you are connected to a remote SSID with the CONNECT command, you can issue DB2 commands to display, start, or stop objects on the remote SSID. To do so, you use the IBM SYSPROC.ADMIN COMMAND_DB2 stored procedure.
Using saved connections

**To install the DB2 stored procedure**

1. Create a JCL startup procedure for the IBM z/OS Workload Manager (WLM) environment.

2. Create the SYSPROC.ADMIN_COMMAND_DB2 stored procedure in the DB2 catalog, and specify the WLM environment.

3. Activate the WLM environment.

For more information, see the IBM *DB2 for z/OS Installation Guide*.

**To connect to a remote SSID and issue the DB2 command**

1. On your local DB2 subsystem, create a database or table space list.

   For more information, see Chapter 2, “Getting started with CATALOG MANAGER.”

2. Connect to a remote SSID.

   For more information, see “Connecting to a specified SSID” on page 166.

3. In the **Cmd** field of the source object, type one of the following commands and press **Enter**:

   - DISPLAY
   - START
   - STOP

   **TIP**
   You can also connect to a remote SSID and issue the DB2 commands in batch. For more information, see “Using the BATCH command for a DB2 object list or a mixed list” on page 207.

**Using saved connections**

When a user makes a connection, CATALOG MANAGER saves it in the *connections list* in the memory of the user’s current session. As the user exits CATALOG MANAGER, the saved connections from the session memory are written to the connections list that is available to all users.

A user can re-establish a saved connection by selecting it from the connections list.
Using saved connections

**NOTE**
The connections list does not duplicate saved connections. Only the first of identical saved connections is displayed on the connections list.

The data on the Connections List panel (Figure 69) includes the following information:

- connections that were saved by all users before you started the current CATALOG MANAGER session
- new, unique connections that you have established during the current CATALOG MANAGER session

**NOTE**
You cannot see new connections made by other users during their current sessions.

---

**Figure 69  Connections List panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Quickname</th>
<th>Location</th>
<th>Collection</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF-S</td>
<td>ACT_QA1010</td>
<td>RDAMSL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFF-DECA</td>
<td>ACT_QA1010</td>
<td>MVSJXL2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFF-DEDK</td>
<td>ACT_QA1010</td>
<td>MVSJXL1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFF</td>
<td>ACT_QA1010</td>
<td>RDAPXB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Select the connection you want with an "S" and press ENTER

---

Bottom of data
Using saved connections

Columns on the Connections List Panel include the following:

- **Quickname** is a user-defined name that identifies a connection on the connections list. If you have SYSADM privileges, you can add or change a quickname by entering the appropriate name in the **Cmd** column of the connection, and then pressing **Enter**.

  Quicknames must be created in the format **SSID-SSID** to be selectable from the Connections List panel or the Connection Selection List panel, or to be used with the QCONNECT command.

- **Location** is the server name of the connection.

- **Collection** is the package set of the connection.

- **Owner** is the person who established the connection first.

**NOTE**

New, unique connections that you establish during the current session are displayed on the connections list without an owner name. The owner name is appended when you exit CATALOG MANAGER.

Re-establishing a saved connection

This procedure describes how to display the connections list and re-establish a saved connection.

1. From the Primary Menu panel, type **QCONNECT (QC)** on the **Command** line.

   The Connections List panel is displayed.

2. In the **Cmd** column of the appropriate row, type **S** to select a connection.

3. Press **Enter** to re-establish the connection and return to the Primary Menu panel.

Using a quickname to re-establish a saved connection

If you know the quickname of a saved connection, you can re-establish the connection while bypassing the Connections List panel.

On the **Command** line of the Primary Menu panel, type **QCONNECT (QC)** followed by the quickname that identifies the connection. For example, enter **QCONNECT DEFF-DECA** to re-establish the connection on row 2 of **Figure 69**.
Viewing the connections table

When CATALOG MANAGER is installed, a connections table is created that includes the DB2 SSID to which CATALOG MANAGER attaches by default. The installer or system administrator can add other SSIDs to make it easier for users to establish attachments and connections.

**NOTE**
For information about how to define or edit DB2 subsystems in the BMCDB2 CLIST, see the installation guide.

1. From the Primary Menu panel, enter CONNECT (CON) on the Command line.

   The Change Access panel is displayed.

2. On the Command line, enter CONTAB.

   The Connections Table panel is displayed (Figure 70).

**Figure 70  Section of Connections Table panel**

<table>
<thead>
<tr>
<th>SSID</th>
<th>Plan</th>
<th>Collection</th>
<th>Nickname</th>
<th>Location</th>
<th>LocID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CK</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>CK</td>
<td></td>
</tr>
<tr>
<td>DEDQ</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDQ</td>
<td></td>
</tr>
<tr>
<td>DEDZ</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDZ</td>
<td></td>
</tr>
<tr>
<td>DEDK</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDK</td>
<td></td>
</tr>
<tr>
<td>DEDV</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDV</td>
<td></td>
</tr>
<tr>
<td>DEDW</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDW</td>
<td></td>
</tr>
<tr>
<td>DECA</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DECA</td>
<td></td>
</tr>
<tr>
<td>DECI</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DECI</td>
<td></td>
</tr>
<tr>
<td>DECS</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DECS</td>
<td></td>
</tr>
<tr>
<td>DEEG</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEEG</td>
<td></td>
</tr>
<tr>
<td>DECC</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DECC</td>
<td></td>
</tr>
<tr>
<td>DEE5</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEE5</td>
<td></td>
</tr>
<tr>
<td>DEEB</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEEB</td>
<td></td>
</tr>
<tr>
<td>DEFU</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEFU</td>
<td></td>
</tr>
<tr>
<td>DEFV</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEFV</td>
<td></td>
</tr>
<tr>
<td>DEEU</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEEU</td>
<td></td>
</tr>
<tr>
<td>DEGA</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEGA</td>
<td></td>
</tr>
</tbody>
</table>
Using DB2-identifiers with the CONNECT command

Table 30 describes the columns on the Connections Table panel.

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSID</td>
<td>lists DB2 subsystems to which CATALOG MANAGER can be attached</td>
</tr>
<tr>
<td>S</td>
<td>indicates how CATALOG MANAGER will access the catalog</td>
</tr>
<tr>
<td></td>
<td>D = direct access to the catalog</td>
</tr>
<tr>
<td></td>
<td>I = indirect access to the catalog by way of a copy or view</td>
</tr>
<tr>
<td></td>
<td>S = either direct or indirect access to the catalog, but CATALOG MANAGER connects to the SSID by way of a server connection</td>
</tr>
<tr>
<td>Plan</td>
<td>lists the main CATALOG MANAGER plans that can be used</td>
</tr>
<tr>
<td>Collection</td>
<td>lists the CATALOG MANAGER collection IDs that can be used</td>
</tr>
<tr>
<td>Nickname</td>
<td>identifies a user-defined name for an attachment or connection</td>
</tr>
<tr>
<td>Location</td>
<td>lists the names of servers that are available for connections</td>
</tr>
<tr>
<td>LocID</td>
<td>lists the SSIDs of servers that are available for connection</td>
</tr>
</tbody>
</table>

The connections table is provided for reference only; you cannot select an attachment or connection from it. However, noting the information that is presented on the connections table can make it easier for you to establish an attachment or connection from the Primary Menu panel or the Change Access panel.

3 After viewing the connections table, press END to display the Change Access panel.

Using DB2-identifiers with the CONNECT command

A DB2-identifier is a value that specifies an attachment or connection listed on the connections table. If you know the DB2-identifier that identifies the connection that you want to establish, you can establish the connection and bypass the Change Access panel by entering the CONNECT command followed by the DB2-identifier on the Command line of the Primary Menu panel.

The types of DB2-identifiers are as follows:

- SSID, a DB2 subsystem for attachment
- server name, the location that is set up in DB2 for a server connection
- server alias, the location ID that is set up in DB2 for a server connection
- collection ID (if one has been created)
- collection ID nickname (if one has been created)
- nickname, a user-defined name that identifies an attachment or connection.
For example, you can enter CONNECT DBDB-DBBF on the Command line of the Primary Menu panel to establish the designated connection. In this example, DBDB-DBBF would be the nickname assigned to the connection.

**Identifying attachments or connections with unique values**

When you enter CONNECT DB2-identifier type of command, CATALOG MANAGER establishes the attachment or connection identified by the first occurrence of the DB2-identifier in the connections table. The values of the DB2-identifiers that you use with the CONNECT command must be unique to prevent unexpected results.

For example, the connections table in Figure 70 on page 173 shows the same SSID value (DBDC) on rows 9 and 10. If you enter CONNECT DBDC, CATALOG MANAGER attempts to attach you to the first instance of DBDC that it finds. However, this instance might not be the SSID and collection that you want.

To prevent such a situation, the installer or system administrator has created unique nicknames for the attachments that use the same SSID. The nicknames are listed in the Nickname column.

You can always verify whether the DB2-identifier that you plan to use is unique by viewing the connections table before entering the command.

**Displaying the current server ID**

The location name and SSID might be the same if you limit the location name to four characters. To eliminate ambiguity when connected to a server, you can set CATALOG MANAGER switches so that all or part of the server name is displayed as you navigate through CATALOG MANAGER.

1. On the Primary Menu panel, in the Action section, type O to select CATALOG MANAGER options processing.

   The Options panel is displayed.

2. In the Edit Switches field, type Y.

   The Switches panel is displayed.

When the value of the Show Server switch is Y, the current server (if any) is displayed in the short message area of alternate CATALOG MANAGER panels.
When the value of the Server SSID switch is Y, the first four characters of the current server (if any) are displayed in the panel ID area of each CATALOG MANAGER panel instead of the SSID.

**Troubleshooting a DB2 Attach or DB2 Connect failure**

If the attempt to attach or connect fails or produces an unexpected result, verify the following conditions:

- The DB2-identifier value that you have used is unique within the connections table or BMCDB2 CLIST.

- CATALOG MANAGER is installed on the remote z/OS at the same maintenance level as on the z/OS from which you are attempting to connect.

- The DDF and values in the DB2 communication database are set up.

**Where to go from here**

The object list is one of the most important features of CATALOG MANAGER. Most CATALOG MANAGER functions can be initiated from an object list. Chapter 5, “Working with lists and searches,” describes how to generate object lists and use them to manage your DB2 catalog information.
This chapter contains the following topics:

Overview ................................................................. 178
Using mixed lists ......................................................... 178
Valid source objects for mixed lists ................................. 178
Generating a mixed list ............................................... 179
Using combined lists ..................................................... 180
Generating a combined list .......................................... 181
Excluding objects from a combined list ............................ 182
Using SEARCH to generate lists based on object attributes .... 183
Valid objects for searches ............................................. 183
Generating a list by using the SEARCH command ............. 185
Using host variables in a search .................................... 190
Using the Quick-Search feature .................................... 192
Using saved search variables in a Quick-Search ............... 193
Using a WHERE clause in a Quick-Search ....................... 194
Including Quick-Searches in BATCH jobs ....................... 194
Creating complex searches ......................................... 194
Creating searches that do not contain a JOIN .................. 195
Creating searches that contain a JOIN ............................ 195
Customizing object list displays .................................... 196
Specifying a new order for displayed columns .................. 197
Sorting a list by one or more columns ............................ 198
Locating string values ............................................... 200
Counting items ............................................................ 200
Describing list objects .................................................. 201
Generating JCL for a job in batch .................................. 206
Using the BATCH command for a DB2 object list or a mixed list 207
Using the BATCH command for a CATALOG MANAGER list or search 210
Generating, editing, and executing SQL ......................... 212
Using Confirm SQL panels .......................................... 212
Working with the SQL_Table ........................................ 213
Applying SQL model statements ................................... 215
Extended SQL processing ........................................... 219
Where to go from here .................................................. 222
Overview

CATALOG MANAGER helps you manage your DB2 catalog with ease through the use of object lists. You can access information deep within catalog tables without creating a series of queries, simply by generating a list of objects, then generating lists of their dependent objects or privileges. CATALOG MANAGER writes the SQL to accomplish these tasks for you, then enables you to save the SQL so that you can use it again.

In Chapter 2, “Getting started with CATALOG MANAGER,” you learned how to generate object lists from the Primary Menu and from other object lists. Now you will learn how to customize list displays and specify qualifiers other than the object name.

Using mixed lists

A mixed list (also called a mixed object list) is a secondary list that shows multiple object types associated with certain source object types. You can generate a mixed list from either a level-one list or a secondary list. A mixed list can display objects that are dependent upon the source object as well as objects upon which the source object is dependent. The object code that generates a mixed list is MX.

**NOTE**

Batch processing is available for the MX command. For information, see “Generating JCL for a job in batch” on page 206.

You cannot use the ALL keyword in a command on the following mixed object lists: AU, FK, IM, MX, PK, TM, and TT.

Valid source objects for mixed lists

Table 31 lists the source objects from which you can generate mixed lists. The instruction area of most list panels specifies whether the MX command is valid for that panel.

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>database</td>
<td>DB</td>
</tr>
<tr>
<td>plan</td>
<td>PL</td>
</tr>
<tr>
<td>storage group</td>
<td>SG</td>
</tr>
<tr>
<td>synonym</td>
<td>SY</td>
</tr>
</tbody>
</table>
Generating a mixed list

This procedure describes how to generate a mixed list of objects that are associated with a table space.

1. Generate a list of table spaces. For information, see “Generating lists in CATALOG MANAGER” on page 47.

2. On the list panel, type MX in the Cmd column beside the table space for which you want to generate a mixed list (Figure 71).

3. Press Enter.

   CATALOG MANAGER displays a mixed list of objects associated with the source table space. Where possible, objects are indented to show dependency (Figure 72).
Using combined lists

Figure 72  Mixed Object List panel generated from table space list

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>02</td>
<td>CSR</td>
</tr>
</tbody>
</table>

CMD will show commands for this list. Type command and press ENTER
Lists: ANY VALID LIST FROM THE SELECTED ROW
QUALIFIER: TABLESPACE=QZUDAC.QZUS01AC

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Obj Type</th>
<th>ObjInf</th>
<th>Tblspace</th>
<th>Name of Object</th>
<th>Plan or Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS</td>
<td>0004</td>
<td>QZUS01AC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.TB</td>
<td>TCPT</td>
<td>QZU.QZUT01_DACS01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.IX</td>
<td>PART</td>
<td>QZU.QZUX01_DACS01T01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.IX</td>
<td>DPSI</td>
<td>QZU.QZUX02_DACS01T01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.IX</td>
<td>PART</td>
<td>QZU.QZUX03_DACS01T01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.IX</td>
<td>DPSI</td>
<td>QZU.QZUX04_DACS01T01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.IX</td>
<td>PART</td>
<td>QZU.QZUX05_DACS01T01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.IX</td>
<td>DPSI</td>
<td>QZU.QZUX06_DACS01T01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.IX</td>
<td>UNIQ</td>
<td>QZU.QZUX07_DACS01T01</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

********************************************************************

NOTE

On the Object Use Options panel (Figure 25 on page 85), you can specify whether to exclude synonyms, plans, and packages from mixed object lists.

You can issue utility and DDL commands against objects on a mixed list as you can on any other list. You can also generate additional secondary lists for objects on a mixed list. For a list of these object types and the lists that you can generate for them, type CMD after generating a mixed object list.

Using combined lists

A combined list displays a single object type that is associated with multiple source objects in a list. To generate a combined list, type objectCode ALL on the Command line. You can generate a combined list from either a level-one list or a secondary list.

Table 32 shows the source object types and codes for which you can generate a combined list.
Generating a combined list

This procedure describes how to generate a combined list of packages that a plan can use.

1. Generate a list of plans. For information, see “Generating lists in CATALOG MANAGER” on page 47.

2. On the Command line, type PI ALL.

3. Press Enter.

The Packlist List panel is displayed, which shows a combined list of all of the packages that the plans use (Figure 73). An asterisk in the Package column indicates that the plan can use all packages in the associated collection.

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>alias</td>
<td>AL</td>
</tr>
<tr>
<td>column</td>
<td>CO</td>
</tr>
<tr>
<td>database</td>
<td>DB</td>
</tr>
<tr>
<td>image copy</td>
<td>IC</td>
</tr>
<tr>
<td>index mixed</td>
<td>IM</td>
</tr>
<tr>
<td>index</td>
<td>IX</td>
</tr>
<tr>
<td>package</td>
<td>PG</td>
</tr>
<tr>
<td>plan</td>
<td>PL</td>
</tr>
<tr>
<td>partition</td>
<td>PT</td>
</tr>
<tr>
<td>relation</td>
<td>RE</td>
</tr>
<tr>
<td>storage group</td>
<td>SG</td>
</tr>
<tr>
<td>table</td>
<td>TB</td>
</tr>
<tr>
<td>view</td>
<td>VW</td>
</tr>
</tbody>
</table>
Figure 73 Combined package list

Excluding objects from a combined list

To exclude some objects in the list of source objects before generating the combined list, perform the following steps:

1. Generate a list of table spaces. For information, see “Generating lists in CATALOG MANAGER” on page 47.

2. Type X in the Cmd column beside the objects that you want to exclude.

3. Press Enter.

The source list panel is displayed. The designated objects are marked as excluded.

4. In the list panel, type the appropriate object type code followed by a space and the keyword ALL on the Command line.

5. Press Enter to generate the combined list.
Using SEARCH to generate lists based on object attributes

In Chapter 2, “Getting started with CATALOG MANAGER,” you learned how to generate lists based on an object name by using the default LIST command and an optional qualifier. However, you can also generate lists that match different and more specific variables by using the SEARCH command.

You can issue a SEARCH command from the Primary Menu panel or a list panel. The basic command format is `SEARCH (SEA) objectCode`. You can enter the SEARCH command in the following ways.

- On the Primary Menu panel or a list panel, on the **Command** line, type `SEARCH (SEA) objectCode`.

- On the Primary Menu panel, use the following values:
  - On the **Command** line, type `SEARCH (SEA)`.
  - In the **Obj type** field, type the object type code or number (if available).

**NOTE**
The SEARCH command uses dynamic SQL. Individual users are required to have SELECT authority on any tables that are referenced when the SEARCH command is issued.

Valid objects for searches

Table 33 lists the source objects for which you can perform searches.

**Table 33  Valid source objects for searches (part 1 of 3)**

<table>
<thead>
<tr>
<th>Object type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>alias</td>
<td>AL</td>
</tr>
<tr>
<td>auxiliary table</td>
<td>XT</td>
</tr>
<tr>
<td>check constraint (for tables created in DB2 Version 7 or later)</td>
<td>C2</td>
</tr>
</tbody>
</table>
Table 33  Valid source objects for searches (part 2 of 3)

<table>
<thead>
<tr>
<th>Object type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>check constraint</td>
<td>CK</td>
</tr>
<tr>
<td>check dependent</td>
<td>CD</td>
</tr>
<tr>
<td>column</td>
<td>CO</td>
</tr>
<tr>
<td>constraint dependent</td>
<td>CP</td>
</tr>
<tr>
<td>data or distinct type</td>
<td>DT</td>
</tr>
<tr>
<td>database</td>
<td>DB</td>
</tr>
<tr>
<td>DBRM</td>
<td>DM</td>
</tr>
<tr>
<td>image copy</td>
<td>IC</td>
</tr>
<tr>
<td>index</td>
<td>IX</td>
</tr>
<tr>
<td>index space</td>
<td>IS</td>
</tr>
<tr>
<td>index space partition</td>
<td>IP</td>
</tr>
<tr>
<td>IP name</td>
<td>IN</td>
</tr>
<tr>
<td>Java archive object</td>
<td>JB</td>
</tr>
<tr>
<td>Java archive contents</td>
<td>JC</td>
</tr>
<tr>
<td>key column use</td>
<td>KU</td>
</tr>
<tr>
<td>locationa</td>
<td>LO</td>
</tr>
<tr>
<td>LU lista</td>
<td>LL</td>
</tr>
<tr>
<td>LU modea</td>
<td>LM</td>
</tr>
<tr>
<td>LU mode selecta</td>
<td>LS</td>
</tr>
<tr>
<td>LU namea</td>
<td>LU</td>
</tr>
<tr>
<td>native SQL procedure</td>
<td>NP</td>
</tr>
<tr>
<td>package</td>
<td>PG</td>
</tr>
<tr>
<td>parameter</td>
<td>FP</td>
</tr>
<tr>
<td>plan</td>
<td>PL</td>
</tr>
<tr>
<td>procedure</td>
<td>PR</td>
</tr>
<tr>
<td>relation</td>
<td>RE</td>
</tr>
<tr>
<td>routine</td>
<td>FN</td>
</tr>
<tr>
<td>routine source</td>
<td>FS</td>
</tr>
<tr>
<td>sequence</td>
<td>SE</td>
</tr>
<tr>
<td>storage group</td>
<td>SG</td>
</tr>
<tr>
<td>string</td>
<td>ST</td>
</tr>
<tr>
<td>synonym</td>
<td>SY</td>
</tr>
<tr>
<td>system user</td>
<td>SU</td>
</tr>
<tr>
<td>table</td>
<td>TB</td>
</tr>
<tr>
<td>table constraint</td>
<td>TC</td>
</tr>
<tr>
<td>table space</td>
<td>TS</td>
</tr>
<tr>
<td>table space partition</td>
<td>TP</td>
</tr>
<tr>
<td>trigger</td>
<td>TR</td>
</tr>
</tbody>
</table>
Generating a list by using the SEARCH command

To generate a list of objects, you set variables by specifying a combination of operators, values, and WHERE clauses. CATALOG MANAGER connects multiple search operators and values with the AND operator and constructs an SQL SELECT statement to fetch objects from the catalog tables.

To generate a list by using the SEARCH command

1. Enter a SEARCH command. For more information, see “Using SEARCH to generate lists based on object attributes” on page 183.

A search panel that is applicable to the specified object type is displayed. The column names that are displayed on the search panel are attributes that are associated with the object type that was specified in the search. Figure 74 shows a search panel for table spaces.

### Table 33  Valid source objects for searches (part 3 of 3)

<table>
<thead>
<tr>
<th>Object type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>US</td>
</tr>
<tr>
<td>user name</td>
<td>UN</td>
</tr>
<tr>
<td>view</td>
<td>VW</td>
</tr>
<tr>
<td>volume</td>
<td>VL</td>
</tr>
</tbody>
</table>

a  Valid if DDF is defined to CATALOG MANAGER.
Generating a list by using the SEARCH command

Figure 74 Search panel for table spaces

```
DEFF-R ----------------------------- SEARCH ------------------------ ROW 1 OF 49
Command ====> Scroll ====> PAGE

Tablespace Search Columns
Save/Retrieve Search * (R-retrieve, S-save)
Edit a WHERE clause N (Y/N)
Column Datatype Length Oper  Value
NAME       VARCHAR    24  =      ''
CREATOR    VARCHAR    128 =      ''
DBNAME     VARCHAR    24  =      ''
OBID       SMALLINT   2   =
OBID       SMALLINT   2   =
PSID       SMALLINT   2   =
BPOOL      CHAR       8   =      ''
PARTITIONS SMALLINT   2   =
LOCKRULE   CHAR       1   =      ''
PGSIZE     SMALLINT   2   =
ERASERULE  CHAR       1   =      ''
STATUS     CHAR       1   =      ''
IMPLICIT   CHAR       1   =      ''
NTABLES    SMALLINT   2   =
NACTIVE    INTEGER    4   =
DSETPASS   VARCHAR    24 =      ''
```

2 In the Oper column, type the operators that indicate the type of comparison that CATALOG MANAGER should use for the search.

Table 34 lists commonly used operators that are valid on a search panel.

Table 34 Valid search operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>selects objects with values equal to the Value field</td>
</tr>
<tr>
<td>&gt;</td>
<td>selects objects with values greater than the Value field</td>
</tr>
<tr>
<td>&lt;</td>
<td>selects objects with values less than the Value field</td>
</tr>
<tr>
<td>LIKE</td>
<td>selects objects with values that match the wildcard pattern entered in the Value field</td>
</tr>
<tr>
<td>^=</td>
<td>selects objects with values that are not equal to the Value field</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>selects objects with values that are not equal to the Value field</td>
</tr>
<tr>
<td>=&gt;</td>
<td>selects objects with values that are not greater than the Value field</td>
</tr>
<tr>
<td>&lt;=</td>
<td>selects objects with values that are not less than the Value field</td>
</tr>
<tr>
<td>=B</td>
<td>selects objects with values that have blank values for this attribute</td>
</tr>
</tbody>
</table>

3 In the Value column, type the values for the fields that CATALOG MANAGER will compare in the search.
4 You can restrict the search even further by creating a WHERE clause in the SELECT statement that CATALOG MANAGER creates for you. To work with the WHERE clause, in the Edit a WHERE clause \( Y \) (Y/N) field, enter Y.

CATALOG MANAGER opens an ISPF edit panel in which you can specify more search criteria than are available on the search panel.

Use the WHERE field for more complex searches, such as the following situations:

- to enter two or more values for the same attribute (the OR operator)
- to use a JOIN operator in the search

When editing the WHERE clause in the WHERE field, observe the following rules:

- Use correct SQL case, punctuation, and syntax as required by DB2.
- Use the per cent (%) and underscore (_) wildcards that are supported by DB2.
- Enclose character strings in quotation marks as required by DB2.
- Use the exact column names as they are displayed on the search panel or in the IBM documentation.

The search variables that were used to generate the list are indicated as a WHERE clause in the instructional area of the panel.

You can also use host variables in the WHERE clause of a search statement. For more information, see “To use host variables to generate a search” on page 190.

5 In the Save/Retrieve Search field, type S or R.

- Type S to save the variables that you use to perform a search. Press Enter.
  
  The Search Options panel (Figure 75) is displayed. Go to step 6.

- Type R to retrieve the saved search variables. Press Enter.
  
  The Search Options panel (Figure 76) is displayed. Go to step 7.

6 In the Save Current Search Variables section (Figure 75), specify values for the search.

The Save Current Search Variables section is already populated with the object type code and object type for your search and with your SQL ID or TSO ID.
Figure 75  Search Options panel - saving search variables

A  In the Owner field, choose one of the following actions:

- To associate the search with a specific session profile, type PROFILE in place of your SQL ID or TSO ID. For more information, see Chapter 11, “Customizing CATALOG MANAGER command access.”

- To specify another user ID as the search owner, type the appropriate user ID in place of your SQL ID or TSO ID.

B  In the Name field, type a name under which you want to save this set of search variables. You can use a maximum of 18 characters for the name.

     **NOTE**
     To associate the search with a specific session profile, type the name of the session profile in the Name field.

C  *(optional)* In the Title field, type a description of the search variables in the set. You can use a maximum of 30 characters in the Title field.

D  In the Edit before saving field, type Y or N to edit the variables before you save them.

E  In the Save current search variables using the following identification field, type Y.
Press Enter.

CATALOG MANAGER saves the search variables and displays the Search panel. A confirmation message is appended to the panel in the short message field.

In the Retrieve Saved Search Variables section (Figure 76), specify identifying values for retrieving saved variables.

Your user ID is the default value in the Owner field.

**Figure 76  Search Options panel - retrieving search variables**

A In the Name field, type the name of the set of saved search variables.

**NOTE**

You can use wildcard characters that are supported by CATALOG MANAGER in both the Owner and Name fields.

B In the Retrieve search variables using the following identification field, type Y.

C In the Edit after retrieval field, type Y or N to edit the search variables after the product retrieves them.
Using host variables in a search

To save time and avoid errors while performing recurrent searches, CATALOG MANAGER enables the use of host variables in the WHERE clause of a search statement. For example, you might need to perform a complex search regularly in which values for some of the variables will change each time that the search is executed.

With CATALOG MANAGER, the search can be created and saved, including host variables for the values that will change. You can retrieve the search as needed, then substitute appropriate values for the host variables before executing the search.

**To use host variables to generate a search**

In the following example, you have created and saved a complex table search that must be performed every day. You will need to specify different values for **CREATOR** and **NAME** each time that you perform the search.

1. Perform a search for tables. For more information, see “Using SEARCH to generate lists based on object attributes” on page 183.

2. On the Search panel, set the applicable variables.
3 In the Edit a WHERE clause field, type Y.

4 Create the WHERE clause in the ISPF edit panel. Define host variables for the CREATOR and NAME fields as in this example:

    CREATOR LIKE :CREATOR AND NAME = :NAME

This syntax indicates that the CREATOR value can include a wildcard character that is supported by DB2.

5 Press Enter.

The SQL Host Variables List panel is displayed (Figure 77).

**Figure 77 SQL Host Variables List panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>SQL Host Variables List</th>
<th>Scroll</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display SQL . . . : N</td>
<td>(Y/N) Display the SQL statement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute SQL . . . : N</td>
<td>(Y/N) Substitute values and execute SQL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Select a variable with 'S' to enter long character values.

<table>
<thead>
<tr>
<th>Host Variable</th>
<th>Substitute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATOR</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td></td>
</tr>
</tbody>
</table>

6 In the Substitute Value column of the CREATOR row, type the value for the CREATOR host variable.

**NOTE**

To enter long host variable values, type S by the host variable name for a full-panel edit.
Table 35 describes the valid host variable values.

Table 35  Host variable values

<table>
<thead>
<tr>
<th>For this value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td>NULL in uppercase</td>
</tr>
<tr>
<td></td>
<td>To specify NULL in the SQL, type NULL in uppercase. For all other values, type a number or character value.</td>
</tr>
<tr>
<td>Numeric</td>
<td>a number without quotation marks</td>
</tr>
<tr>
<td></td>
<td>Valid numeric values are not placed in quotation marks.</td>
</tr>
<tr>
<td>Character</td>
<td>characters with or without quotation marks</td>
</tr>
<tr>
<td></td>
<td>Characters that are entered without quotation marks are quoted and translated to uppercase. Characters that are entered in quotation marks are used as specified and are not translated to uppercase.</td>
</tr>
</tbody>
</table>

7 In the Substitute Value column of the NAME row, type the value for the NAME host variable.

8 Choose one of the following actions:

- To see the SELECT statement that CATALOG MANAGER creates, in the Display SQL field, type Y. Press Enter.

  The SQL Statement Display panel is displayed. You can only view the SQL statement.

- To execute the SELECT statement after specifying the substitute values, in the Execute SQL field, type Y. Press Enter.

  CATALOG MANAGER performs the search.

  If you type N in the Execute SQL field, CATALOG MANAGER displays the SQL Host Variables List panel, from which you can continue to make changes to the substitute values. CATALOG MANAGER does not perform the search until you type Y in the Execute SQL field and press Enter.

Using the Quick-Search feature

The Quick-Search feature of CATALOG MANAGER enables you to save time by performing searches that use saved search variables or WHERE clauses with fewer steps. You enter all of the information that CATALOG MANAGER needs to perform the search on the Command line of the Primary Menu panel or a list panel. You can enter a maximum of 48 characters.
Using saved search variables in a Quick-Search

For searches that involve saved search variables, follow this general procedure:

1. On the **Command** line of the Primary Menu panel or a list panel, type the following:

   - SEARCH (SEA)
   - a space
   - the two-character object type code
   - a space
   - the name of the saved search variable set

   **NOTE**
   CATALOG MANAGER assumes the current set SQL ID unless you specify another owner. You can also use a wildcard in the entry.

2. Press **Enter**. The results are as follows:

   - If only one set of search variables has been saved under the specified name, CATALOG MANAGER performs the search and displays the matching items in a list.

     For example, entering **SEA DB MMS2.MMSTEST** displays a list that uses the search variables saved in MMSTEST by the SQL ID MMS2.

   - If more than one set of criteria is found, CATALOG MANAGER displays a selection list. You can select the appropriate set of search variables and perform the search.

     For example, entering **SEA TB * displays a selection list of saved variables for table searches saved by the current set SQL ID.**
Using a WHERE clause in a Quick-Search

For quick-searches that include WHERE clauses, follow this general procedure:

1. On the Command line of the Primary Menu panel or a list panel, type the following:
   - SEARCH (SEA)
   - a space
   - the two-character object type code
   - a space
   - an abbreviated version of the WHERE clause, omitting the WHERE operator, and using only correct SQL case, punctuation, syntax, and wildcards that are acceptable to DB2.

2. Press Enter.

   For example, enter SEA DB NAME LIKE ‘DEMO%’ to display a list of databases with names that begin with DEMO.

Including Quick-Searches in BATCH jobs

BATCH jobs can include the SEARCH command. The batch SYSIN does not generate the command; you must manually add the command to the input stream. For information, see “Using the BATCH command for a CATALOG MANAGER list or search” on page 210.

Creating complex searches

With CATALOG MANAGER, you can perform, name, save, and retrieve complex searches, such as searches that include JOIN subqueries. These searches, however, require that you have detailed knowledge of SQL and the DB2 catalog tables and their interrelationships.
Creating searches that do not contain a JOIN

To perform a complex search other than a JOIN, follow this general procedure:

1. Perform a search for an object. For more information, see “Using SEARCH to generate lists based on object attributes” on page 183.

   The Search panel is displayed.

2. On the Search panel, type the appropriate values for the displayed Column fields.

3. In the Edit a WHERE clause field, type Y.

   An ISPF edit panel is displayed.

4. Type an SQL statement that includes the necessary operators and values to perform the search.

   **NOTE**
   Remember that you must use correct SQL case, punctuation, syntax, and wildcards that are acceptable to DB2 whenever you use the WHERE field.

Creating searches that contain a JOIN

The object search panel in CATALOG MANAGER enables you to create searches that require a JOIN among multiple catalog tables. As with any other search, you can name, save, and retrieve the search variables.

The procedure for a joined search differs from the procedure described for other types of complex searches. Observe the following general rules when creating a search that includes a JOIN:

1. Perform a search for an object. For more information, see “Using SEARCH to generate lists based on object attributes” on page 183.

   The Search panel is displayed.

2. On the Search panel, do not enter any values in the Column Name fields that see the object attributes.

3. In the Edit a WHERE clause field, enter Y.
Customizing object list displays

4 Complete the WHERE field as follows:

A Type a comma (,) as the first character.

B Type the qualified names of the additional catalog tables to be included, separated by commas.

NOTE
Because some tables have identical column names and the column names in a join must be unique, type a correlation identifier after each qualified table name. For example:

, SYSIBM.SYSTABLES B

C Type the WHERE operator and WHERE clause. For example:

WHERE A.DBNAME IN ('DSNDB01','DSNDB06')
AND A.IBMREQD <> 'Y'
AND A.TBNAME = B.NAME
AND A.TBCREATOR = B.CREATOR
AND B.TSNAME NOT IN ('SYSPKAGE','SYSPLAN')

NOTE
Remember that you must use correct SQL case, punctuation, syntax, and wildcards that are acceptable to DB2 whenever you use the WHERE clause. For the default object, use a correlation identifier of A.

Customizing object list displays

To view all of the information on an object list display, scroll left and right to see all columns, and up and down to see all retrieved rows. Standard scrolling commands, such as increments, minimum, and maximum, are valid.

Because you might not need to see all of the column information regularly, CATALOG MANAGER enables you to customize object list displays by

- specifying which columns are displayed when you generate a list
- specifying the order of columns that are displayed

The specifications are saved in your ISPF profile.

You can also locate specific string values, print lists, and display data in a dump format.
For detailed information about one or more catalog table columns, press HELP after generating the object list.

## Specifying a new order for displayed columns

This procedure describes how to reorganize the columns of a table space list so that the columns appear in a different order.

1. **Generate a list of table spaces.** For information, see “Generating lists in CATALOG MANAGER” on page 47.

2. **On the Command line, type ORDER.**

3. **Press Enter.**

   CATALOG MANAGER displays a list of the columns that are available for the table spaces (Figure 78).

### Figure 78 Column Order Specification panel

```plaintext
DEFF-R------------- Column Order Specification --------- ROW 1 OF 47
Command ===> Scroll ===> PAGE 02

Specify sequence numbers to place columns in desired order, then press END
Type SAVE command to save the selected order as the list default.
Order Colno Column Name
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
1   TRADITIONAL LIST LINE FORMAT
2   1 NAME
3   2 CREATOR
4   3 DBNAME
5   4 DBID
6   5 OBID
7   6 PSID
8   7 BPOOL
9   8 PARTITIONS
10  9 LOCKRULE
11 10 PGSIZE
12 11 ERASERULE
13 12 STATUS
14 13 IMPLICIT
15 14 NTABLES
16 15 NACTIVE
17 16 DSETPASS
```

4. **Specify a new sequence number for each of the columns that you want to re-order.**
After you view or change the values, you can continue by using one of the following methods:

- Press END to save the changes. The values are stored in your profile for use in the current session until you change them again.

- Type SAVE on the Command line and press Enter to save the changes. The values are stored in your profile for use in the current and future sessions until you change them again.

**Sorting a list by one or more columns**

This procedure describes how to sort a list by one or more of the columns of the DB2 table. When using customizable lists you can sort on any columns of the table, even if the columns are not currently displayed.

CATALOG MANAGER retrieves data that can use any encoding scheme from the DB2 catalog. When CATALOG MANAGER executes SQL that uses an ORDER BY clause against the DB2 catalog, the following behavior occurs:

- *(DB2 Version 7)* The query uses an EBCDIC collating sequence to sort data, and CATALOG MANAGER uses the same sequence to display the data on panels and in reports.

- *(DB2 Version 8 and later)* The query uses a Unicode collating sequence to sort data and typically uses the same sequence to display the data on panels and in reports. However, when CATALOG MANAGER sorts the data retrieved from the DB2 catalog queries, the product displays sorted data on panels and in reports in an EBCDIC collating sequence.

1. Generate a list of table spaces.

   For information, see “Generating lists in CATALOG MANAGER” on page 47.

2. On the Command line, type SORT.

3. Press Enter.

   CATALOG MANAGER displays a list of the columns that are available for the sorting (Figure 79).
In the **SEQUENCE** column, specify a sort order for the columns. You can sort from 1 to 9 columns.

In the **ASC/DESC** column, type **A** or **D** for any of the columns that you want displayed in ascending or descending order.

After you view or change the values, you can continue by using one of the following methods:

- Press END to save the changes. The values are stored in your profile for use in the current session until you change them again.

- Type **SAVE** on the **Command** line and press **Enter** to save the changes. The values are stored in your profile for use in the current and future sessions until you change them again.
Locating string values

You can locate a string value in the rows that CATALOG MANAGER displays in the object list by using the FIND or RFIND commands.

FIND locates a character string that you specify in the command. RFIND locates the next occurrence of the character string that is specified in the FIND command. For more information, see “Command-line commands” on page 470.

Counting items

The COUNT command, entered in the **Cmd** (C) column of a list panel beside a valid object type, displays the number of rows in the source object (Figure 80). The COUNT command is valid with table spaces, tables, views, aliases, and synonyms.

Figure 80  Table Count List panel

![Table Count List panel]

CATALOG MANAGER also provides catalog count statistics that show the number of a given object type in the catalog. To display catalog count statistics, enter the CATSTATS command on the **Command** line of the Primary Menu panel or an object list panel. The DB2 Catalog Counts panel is displayed (Figure 81).
Describing list objects

CATALOG MANAGER provides the following commands for displaying object descriptions:

- DESCRIBE
- DES
- D
- S
- DESTATISTICS
- HDESC

Each command meets a specific need. You can print the descriptive information if a hardcopy is needed.
**DESCRIBE command**

To generate a detailed description of a list object, type `DESCRIBE` in the `Cmd` column next to the source object.

The DESCRIBE command displays detailed information that is stored in the DB2 catalog about a specific object, including structure and dependencies. Figure 82 shows a sample report that is displayed when the DESCRIBE command is applied to a table. Information in the report is dependent upon the source object type. You can specify your options for displaying the report for plans and packages on the Describe Options panel. (For more information, see “Setting DESCRIBE options” on page 98.)

**NOTE**

In the Describe Database panel, CATALOG MANAGER displays the number of partitions and the segment size for a table space in the `ObjInfo` column.

The DESCRIBE command also displays detailed information about CATALOG MANAGER log entries for the Audit, Session, and Drop Recovery Logs.
Table: QZU.QZUT01_DEBS01

<table>
<thead>
<tr>
<th>Command ===</th>
<th>Table: QZU.QZUT01_DEBS01</th>
<th>Scroll ===</th>
<th>PAGE</th>
</tr>
</thead>
</table>

Table = QZU.QZUT01_DEBS01
FROM SYSIBM.SYSTABLES

---

**Creator**... QZU  **Type**... T
**Table Name**... QZUT01_DEBS01  **Status**... X
**Database**... QZUDEB  **Checkflag**...
**Tablespace**... QZUS01EB  **Checkrid**... X'40404040'
**Editproc**...  **Validproc**...
**Audit**...  **Parents**...
**Pctpages**... 75  **Children**...
**Colcount**... 12  **DBID**... 1204
**Record Length**... 86  **OBID**... 3
**Key Columns**... 5  **Key OBID**... 6
**Createdby**... RDABKH1  **Label**...
**Remarks**...  **Createdts**... 06-10.40.20.849236
**Alteredts**... 06-10.40.20.849236  **Datacapture**...
**Rb1**... X'006009FB9000'  **Rba2**... X'006009FB9000'
**Pctrowcomp**... 70  **Statsime**... 07-09.12.32.710646
**Orphan**...  **Clustertype**...
**IBM reqd**... I  **Checks**...
**Viewdeps**...  **Cardf**... 988013
**Checkrid5b**... X'4040404040'  **Encoding scheme**... E
**Tbcreator**...  **Tbname**...
**Npagesf**... 14768  **Spacef**... 79920
**Avgrowlen**... 60  **Relcreated**...
**Tablestatus**... 0  **Number dep MQTs**...
**Version**... 0  **Partkeycolnum**...
**Split Rows**...  **Security Label**...
**Owner**... QZU  **Append**... N
**Ownertype**...
---

**TABLE CONSTRAINTS**

<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Type</th>
<th>Enforcing Index</th>
<th>Creator</th>
<th>Colcount</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>P</td>
<td>QZU.QZUX02_DEBS01T01</td>
<td>RDABKH1</td>
<td>5</td>
</tr>
</tbody>
</table>

**TABLE COLUMNS**

<table>
<thead>
<tr>
<th>Num</th>
<th>Column Name</th>
<th>Coltype</th>
<th>Length</th>
<th>Null</th>
<th>Up</th>
<th>Pkey</th>
<th>Dfl</th>
<th>Bit</th>
<th>Length2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DATE</td>
<td>DATE</td>
<td>4</td>
<td>N</td>
<td>Y</td>
<td>1</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>AUTHID</td>
<td>CHAR</td>
<td>8</td>
<td>N</td>
<td>Y</td>
<td>2</td>
<td>N</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TBCREATOR</td>
<td>CHAR</td>
<td>8</td>
<td>N</td>
<td>Y</td>
<td>3</td>
<td>N</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TBNAME</td>
<td>CHAR</td>
<td>18</td>
<td>N</td>
<td>Y</td>
<td>4</td>
<td>N</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PLAN</td>
<td>CHAR</td>
<td>8</td>
<td>N</td>
<td>Y</td>
<td>5</td>
<td>N</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>LAST_MNT_DATE_TIME</td>
<td>TIMESTAMP</td>
<td>10</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>NTRANS</td>
<td>INTEGER</td>
<td>4</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>INSERTS</td>
<td>INTEGER</td>
<td>4</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>UPDATES</td>
<td>INTEGER</td>
<td>4</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DELETES</td>
<td>INTEGER</td>
<td>4</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>MIN_TIME</td>
<td>TIME</td>
<td>3</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MAX_TIME</td>
<td>TIME</td>
<td>3</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Row length - maximum 86 - minimum 86 including eight byte header

---

END OF DATA
**DES command**

The DES command, for some object types, displays a subset of the description that is provided by the DESCRIBE command. The affected object types and descriptive information are shown in Table 36.

**Table 36  DES command descriptions**

<table>
<thead>
<tr>
<th>Object types</th>
<th>Excluded information</th>
</tr>
</thead>
<tbody>
<tr>
<td>table</td>
<td>indexes, foreign keys, table partition columns, table partition limit keys, and materialized query table (MQT) text</td>
</tr>
<tr>
<td>database</td>
<td>plans</td>
</tr>
<tr>
<td>DBRM package</td>
<td>Explain access information from the PLAN_TABLE</td>
</tr>
</tbody>
</table>

For object types that are not shown in Table 36, DES and DESCRIBE display the same information.

**D and S commands**

A partial column value is displayed when the width of the column is larger than the width of the terminal. To see the entire value for a column in a row, you can use the S or D command to display a detailed panel for the rows.

**NOTE**

In CATALOG MANAGER, the D command is equivalent to the SELECT (S) command.

**DESTATISTICS command**

The DESTATISTICS command displays the catalog row and associated statistics for a source table space, table, or index. Figure 83 shows the Describe Table panel generated from the DESTATISTICS command for the example table that is used in Figure 82.
Figure 83  Describe Table panel generated from DESTATISTICS command

Table: QZU.QZUT01_DEBS01

Command ===>                          Scroll ===> PAGE

Table = QZU.QZUT01_DEBS01
-----------------------------------------------
FROM SYSIBM.SYSTABLES

Creator. . . QZU                Type . . . T
Table Name . QZUT01_DEBS01    Status . . . X
Database . QZUDEB                Checkflag.
Tablespace . QZUS01EB         Checkrld. . X'40404040'
Editproc. . . Validproc.
Audit. . . . . . Parents. . . 0
Pctpages . 75                 Children . . 0
Colcount . 12                DBID . . . 1204
Record Length . 86           OBID . . . 3
Key Columns . 5               Key OBID . . 6
Createdby. . . RDABKH1          Label. . .
Remarks. . . . . . . . . . . . . 06-10.40.20.849236
Alteredts. . . 06-10.40.20.849236
Rba1 . . X'006009FB9000'       Rba2 . . X'006009FB9000'
Pctrowcomp . 70                Statstime. . . 07-09.12.32.710646
Orphan. . . Clustertype.
IBM reqd . . . Checks . . 0
Viewdeps . . Cardf. . . . 988013
Checkrld5b . X'4040404040'    Encoding scheme. E
Tbcreator. . . Tbname . .
Npagesf . 14768             Spacef . . . 79920
Avgrowlen. . 60              Relcreated . . 0
Tablestatus . . . Number dep MQTs. 0
Version . . . 0               Partkeycolnum. . 0
Owner. . . . QZU              Append . . . N
Ownerntype . .

--- TABLE COLUMN STATISTICS ---

<table>
<thead>
<tr>
<th>Num</th>
<th>Column Name</th>
<th>Coltype</th>
<th>Length</th>
<th>Key</th>
<th>High2key</th>
<th>Low2key</th>
<th>Cardf</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DATE</td>
<td>DATE</td>
<td>4</td>
<td>1</td>
<td>..</td>
<td>..</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>200122222222</td>
<td>160122222222</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>054500000000</td>
<td>903800000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TSSNOTES.... ANDPDONR .... 18</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EEDEEECE2222 CECDOD42222</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>322563520000</td>
<td>164765900000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UZU .... QZU .... 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EEE444442222</td>
<td>DEE444442222</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>AUTHID</td>
<td>CHAR</td>
<td>8</td>
<td>2</td>
<td>T</td>
<td>W</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EEDEEECE2222 CECDOD42222</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>322563520000</td>
<td>164765900000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UZU .... QZU .... 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EEE444442222</td>
<td>DEE444442222</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TBCREATOR</td>
<td>CHAR</td>
<td>8</td>
<td>3</td>
<td>..</td>
<td>..</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4940000000000 B94000000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>UZU .... QZU .... 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EEE444442222</td>
<td>DEE444442222</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TBNAME</td>
<td>CHAR</td>
<td>18</td>
<td>4</td>
<td>ZZZ</td>
<td>A&amp;P</td>
<td>204</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EEE444444444 C5D44444444</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>999000000000</td>
<td>187000000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WITCP03 .... $A486</td>
<td>295</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ECEEDFF42222 5CFF4442222</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>693370300000</td>
<td>B1486000000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PLAN</td>
<td>CHAR</td>
<td>8</td>
<td>5</td>
<td>..</td>
<td>..</td>
<td>1760</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>020020482722 19000224224224</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>058425307100</td>
<td>996521728700</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>LAST_MNT_DATE_TIME</td>
<td>TIMESTAMP</td>
<td>10</td>
<td>..</td>
<td>F</td>
<td>..</td>
<td>3168</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B0C2222222222</td>
<td>800022222222</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>085300000000</td>
<td>000200000000</td>
<td></td>
</tr>
</tbody>
</table>
**HDESCRIBE command**

The HDESCRIBE (HDESC) command displays a series of panels that show descriptions, starting with the source object and moving to dependent objects. Press END to move from one description panel to the next.

**Printing lists**

The PRINT (PRI) command produces different results depending on whether the command is used on the **Command** line or in the **Cmd** column of a list:

- To write the contents of an entire list to your print data set, enter PRI on the **Command** line of the list panel.

- To generate a DESCRIBE report and write it to your print data set, enter PRI in the **Cmd** column of a list.

The prefix PRI distinguishes the CATALOG MANAGER PRINT command from the PRINT command of ISPF. You can also use your specified command recognition character with the PRINT command (see “Using a command recognition character” on page 37).

**Generating JCL for a job in batch**

For extensive lists and searches, you can use the BATCH command to generate JCL so that you can run commands in batch mode. For DB2 object lists, CATALOG MANAGER includes the list commands in the SYSIN input stream of the JCL. For CATALOG MANAGER lists and searches, you can insert the commands manually into the input stream.

CATALOG MANAGER provides output from the batch JCL job in the SYSPRINT and ACTPRINT data sets for all of the commands. For the DDL BATCH and HDDL BATCH commands, CATALOG MANAGER also provides output (SQL) in a sequential data set. For the HDDL BATCH command on a database object, CATALOG MANAGER also provides SQL in a partitioned data set.
Using the BATCH command for a DB2 object list or a mixed list

1. Generate a list:
   - To generate an object list, see “Generating lists in CATALOG MANAGER” on page 47.
   - To generate a mixed list, see “Generating a mixed list” on page 179.

2. If you generated an object list in step 1, type `command BATCH` or `DB2command BATCH` in the `Cmd` column of the source object, and press Enter.

   The variable `command` is one of the following commands:
   - CONNECT (can only be issued from a location, or LO, list)
   - DCL
   - DDL
   - DES
   - DESCRIBE
   - DESTATISTICS
   - HDDL
   - HDESCRIBE
   - HGRANT

   The variable `DB2command` is one of the following commands:
   - DISPLAY
   - START
   - STOP

   CATALOG MANAGER displays the following message:

   BMC14651 Use BATCH command to generate JCL for all saved commands.

3. In the `Cmd` column of the source object or on the `Command` line, type `BATCH` and press Enter.

   The CATALOG MANAGER Batch Job panel is displayed (Figure 84).
Using the \texttt{BATCH} command for a DB2 object list or a mixed list

4 In the \textbf{JCL Dataset} field, specify the member name of a partitioned data set.

This field indicates the name of the data set in which the generated JCL will be stored.

5 In the \textbf{Set JCL options} field, type \texttt{Y} to view the JCL Generation Options panel, from which you can specify the values for options that control operations.

6 When the JCL Generation Options panel is displayed, press END to return to the CATALOG MANAGER Batch Job panel.

7 In the \textbf{Build Job} field, type \texttt{Y} to create the JCL and save it in the specified JCL data set.

8 In the \textbf{Edit Dataset} field, type \texttt{Y} to edit the JCL data set.

\textit{NOTE}

The I parameter in the ISPSTART command in the SYSTSIN DD statement identifies whether you are connected to an indirect catalog (I=YES) or a direct (real) catalog (I=NO). The collection ID for indirect access is obtained from the installation options module.

\begin{verbatim}
//SYSTSIN DD *
ISPSTART PGM(ACTBMAIN),PARM(0=DC91QEDK,S=DEDK,I=YES,V=DEDKCAT)
\end{verbatim}
9 In the Job Options section of the panel, specify your options for the batch processor:

--- NOTE ---
Ensure that the Ssid and Default options fields specify values for the local DB2 SSID.

A To specify a different SSID to which the batch processor will connect, type the name of the SSID.

The value displayed for the SSID field is the current SSID, not the value saved in the ISPF profile.

B To specify the installation options module, type the name of the module in the Default options field.

The value displayed in the Default options field is the current installation options module, not the value saved in the ISPF profile.

C (DCL, DDL, HDDL, and HGRANT commands) To specify the name of the sequential data set for output, in the HDDL output dsn field, type the name.

If this data set does not exist, CATALOG MANAGER prompts you to allocate it when you build the job.

10 Press Enter.

The generated JCL is displayed in an ISPF edit session.

11 (optional for the CONNECT command) Modify the CONNECT statement in the SYSIN input stream to add the CONNECT command parameters.

The command parameters are defined in Table 29 on page 167.

--- WARNING ---
BMC strongly recommends that you exercise caution when modifying the commands in the SYSIN input stream. Using the incorrect syntax could produce unpredictable results.
12 *(optional for the DB2 DISPLAY, START, or STOP command)* Modify the DB2 commands in the SYSIN input stream to specify objects.

**WARNING**

BMC strongly recommends that you exercise caution when modifying the commands in the SYSIN input stream. Using the incorrect syntax could produce unpredictable results.

Table 37 describes the command syntax.

**Table 37  DB2 command syntax**

<table>
<thead>
<tr>
<th>Command syntax</th>
<th>Valid object lists</th>
<th>Valid object types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY objectType objectName</td>
<td>DB, TS</td>
<td>DB, TS, FN&lt;sup&gt;a&lt;/sup&gt;, PR&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>START objectType objectName</td>
<td>DB, TS</td>
<td>DB, TS, FN&lt;sup&gt;ab&lt;/sup&gt;, PR&lt;sup&gt;ab&lt;/sup&gt;</td>
</tr>
<tr>
<td>STOP objectType objectName</td>
<td></td>
<td>DB, TS, FN&lt;sup&gt;ab&lt;/sup&gt;, PR&lt;sup&gt;ab&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

<sup>a</sup> Part 1 of objectName cannot exceed 8 characters, and part 2 cannot exceed 18 characters.

<sup>b</sup> Because of a DB2 restriction, you cannot issue the command for this object on a remote DB2 SSID.

13 *(optional for the HDDL command)* For a database list, specify the name of a partitioned data set for output by adding PDS=DataSetName to the SYSIN input stream.

14 Press END to return to the CATALOG MANAGER Batch Job panel.

15 In the Submit field, type Y to submit the JCL.

**Using the BATCH command for a CATALOG MANAGER list or search**

1 On the Command line of a Primary Menu panel or a list panel, type BATCH.

2 Press Enter.

   The CATALOG MANAGER Batch Job panel is displayed.

3 In the Build Job field, type Y to create the JCL.

4 Near the bottom of the JCL, find the NO CATALOG MANAGER COMMANDS message in the SYSIN stream *(Figure 85)*.
**Figure 85**  NO CATALOG MANAGER COMMANDS message in JCL

<table>
<thead>
<tr>
<th>ISREDDE2</th>
<th>RDACRJ2.BMCCAT.JCL</th>
<th>Columns 00001 00072</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td></td>
<td>Scroll =&gt; CSR</td>
</tr>
<tr>
<td>000098</td>
<td>//SYSIN DD *</td>
<td></td>
</tr>
<tr>
<td>000099</td>
<td>DSN=RDACRJ2.DEDK.HDDL</td>
<td></td>
</tr>
<tr>
<td>000100</td>
<td>NO CATALOG MANAGER COMMANDS</td>
<td></td>
</tr>
<tr>
<td>000101</td>
<td>/<em>-------------------------------------------------</em>/</td>
<td></td>
</tr>
<tr>
<td>000102</td>
<td>/* END OF JOBSTEP</td>
<td></td>
</tr>
<tr>
<td>000103</td>
<td>/<em>-------------------------------------------------</em>/</td>
<td></td>
</tr>
<tr>
<td>000104</td>
<td>/<em>-------------------------------------------------</em>/</td>
<td></td>
</tr>
<tr>
<td>000105</td>
<td>/* END OF JOB</td>
<td></td>
</tr>
<tr>
<td>000106</td>
<td>/<em>-------------------------------------------------</em>/</td>
<td></td>
</tr>
<tr>
<td>******</td>
<td>*************************************** Bottom of Data ***************************************</td>
<td></td>
</tr>
</tbody>
</table>

5 Type over the message with one or more of the following commands (each on a separate line):

- **DOPTS**
- **DSNZPARAM**
- **ENVI**

If you are connected to a remote DB2 SSID, the ENVI command does not display the following information:

- CATALOG MANAGER plans
- collections other than the default collection
- CATALOG MANAGER table synonyms
- DB2 catalog synonyms

- **LIST**  *type*  *qualifier*

The variable *type* is a two-character object type, and *qualifier* is a character string.

- **LIST MX**  *type*  *qualifier*

The variable *type* is a two-character object type (DB, FK, PG, PL, PR, SG, TB, TR, TS, TT, XT, or VW), and *qualifier* is a fully qualified name.
Generating, editing, and executing SQL

- **SEARCH type savedSearchName**

  The variable *type* is a two-character object type, and *savedSearchName* is the qualified name for the search variables.

- **SEARCH type whereExpression**

  The variable *type* is a two-character object type, and *whereExpression* includes the search variables for a WHERE clause.

**NOTE**

Do not include the WHERE operator in the *whereExpression*. The maximum number of characters permitted is 48.

6 Press END to return to the CATALOG MANAGER Batch Job panel.

7 In the **Submit** field, type **Y** to submit the JCL.

**Generating, editing, and executing SQL**

CATALOG MANAGER generates the necessary SQL statements to perform your catalog management tasks. You can edit, save, and execute the SQL according to your particular needs. CATALOG MANAGER provides the Confirm SQL panel and the SQL_Table List panel for these purposes.

**Using Confirm SQL panels**

Use the Confirm SQL panel to perform the following functions:

- edit SQL command options
- edit the generated SQL
- name and save the generated SQL statements
- execute the SQL

Figure 86 shows the Confirm SQL panel that is displayed when you create a table. The fields on Confirm SQL panels are dependent upon the operation that you are performing. You set the initial values for the Confirm SQL panel by using the SQL and Confirm Options panel (see “Setting SQL and confirm options” on page 90).
When you save SQL, it is stored in the SQL_Table. This table can store any SQL generated during a CATALOG MANAGER session as well as SQL from a PDS or sequential file outside the SQL_Table.

### Displaying the SQL_Table list

You can display saved SQL of which you are the owner or SQL that is owned by all users, as follows.

- To display the SQL_Table List that includes only SQL of which you are the owner, type **SQL** on the **Command** line of the Primary Menu panel or a list panel, and then press **Enter**.

- To display the SQL_Table List for all users, type **SQL *.*** on the **Command** line of the Primary Menu Panel or a list panel, and then press **Enter**.

**Figure 87** shows a section of a sample SQL_Table List panel.
Copying SQL in the SQL Table

This procedure describes how to copy SQL that is in the SQL Table.

1 Display the SQL Table list. For more information, see “Displaying the SQL Table list” on page 213.

2 In the Cmd column beside the entry to be copied, type EDIT newName.

newName can be in the format authID.sqlname, where authID is optional. If you omit authID, your authorization ID is automatically supplied as the owner.

3 Press Enter to display an ISPF edit panel.

NOTE
If the work data set has not been allocated, the Allocate Data Set panel is displayed. To allocate the data set, see “Defining an options data set” on page 77.

4 Edit the SQL as necessary.

5 Press END to save the SQL.


### Copying external SQL

This procedure describes how to copy SQL that has a similar LRECL from a PDS or sequential file from a source that is outside the SQL_Table.

1. From the Primary Menu panel, type **SQL EDIT newName** on the **Command** line.

2. Press **Enter**.

   An ISPF edit panel is displayed.

3. On the **Command** line, type **COPY**.

4. Press **Enter**.

   The ISPF Edit/View-Copy panel is displayed.

5. Specify the library member or sequential data set that you want to copy.

6. Press **Enter**.

   The member or sequential data set is displayed in the ISPF edit panel with **Member member-name copied** in the short message area of the panel.

7. Edit the member, or press **END** to save the SQL.

### Applying SQL model statements

From any list panel, you can apply an SQL model statement that contains host variables to the list objects by using the APPLY command. For the host variables in the SQL model statement, CATALOG MANAGER substitutes values in the DB2 catalog row indicated by the list entry and creates one statement for each object in the list. The host variables must be DB2 catalog column names.

By using the APPLY command, you can significantly reduce the time required to perform the same action against a group of objects. For example, using one model statement, you can add the RESTRICT ON DROP attribute to a group of tables.

#### To apply SQL statements to lists

The following procedure combines an SQL model statement with a list of tables to generate SQL with host variables that adds the RESTRICT ON DROP attribute to a group of tables.
1 Create an SQL model statement in the CATALOG MANAGER SQL_Table.

For this example, create a member called RESTRICT, which contains two host variables, :CREATOR and :NAME. The host variables represent the DB2 catalog columns that contain the data to be substituted for the variables. (If you omit the semicolon (;) at the end of the SQL model statement, CATALOG MANAGER supplies it.)

2 Generate an object list. For information, see “Generating lists in CATALOG MANAGER” on page 47.

3 On the object list, in the Cmd column, type X beside objects that you want to exclude from processing.

4 Press Enter.

5 On the Command line, type APPLY memberName ALL.

In this example, the SQL_Table memberName is RESTRICT.

6 Press Enter.

CATALOG MANAGER displays the Confirm APPLY SQL MODEL panel, which contains the SQL model statement at the bottom of the panel (Figure 88).

**Figure 88 Confirm APPLY SQL MODEL panel**

```
DEFF-R -------------------  Confirm APPLY SQL MODEL  -------------------------
Command ===> 
Edit SQL Model .... N (Y/N)  Current SQLID : RDACRJ2
Replace with SQL Member .
Save model in SQL table . N (Y/N)
Apply to list objects . . N (Y/N)
Name of saved SQL ..... RDACRJ2,RESTRICT
------------------------------------------------------------------- SQL -------------------------------
More: +
ALTER TABLE :CREATOR :NAME ADD RESTRICT ON DROP;
************************************************************************** BOTTOM OF DATA **********************
```
In the Confirm APPLY SQL MODEL panel, you can edit and save the SQL, and apply the SQL to the objects in the list.

**A** In the **Edit SQL Model** field, type Y or N to edit the SQL. Then, press **Enter**.

An ISPF edit panel is displayed, from which you can edit the SQL. Press END to return to the Confirm APPLY SQL MODEL panel.

**B** In the **Replace with SQL Member** field, type the name of another member to replace the existing member. Then, press **Enter**.

**C** In the **Save model in SQL table** field, type Y or N to save the SQL. Then, press **Enter**.

**D** In the **Name of saved SQL** field, specify the owner and name for the SQL. Then, press **Enter**.

**E** In the **Apply to list objects** field, type Y or N to apply the SQL model statement to list objects that were not excluded. Then, press **Enter**.

CATALOG MANAGER displays the Confirm SQL panel (Figure 89). Values from the DB2 catalog have been substituted for the host variables.

**Figure 89** Confirm SQL panel

```
DEFF-R --------------------------  Confirm SQL  ------------ Row 1 to 15 of 235
Command ===>                                                  Scroll ===> PAGE 02
Actions  Edit options . . . .  N  (Y/N)
         Edit SQL . . . . . .  N  (Y/N)  Current SQLID : RDACRJ2
         Save SQL . . . . . .  N  (A/R/Y/N A-append, R-replace, Y-append)
         Execute . . . . . .  N  (Y/N)
Options  Name of saved SQL     20110209_325680
         ------------------------------------  SQL  ------------------------------------
         ALTER TABLE QZU . QZUT01_DCI08S01
         ALTER TABLE QZU . QZUT02_DCI08S01
         ALTER TABLE QZU . QZUT03_DCI08S01
```

Chapter 5 Working with lists and searches
On the Confirm SQL panel, you can edit and save the SQL and then execute it.

A (optional) From the Command line, issue the SET sqlid command to change the value of the current SQLID.

**NOTE**
The ID shown in the Current SQLID field must have the proper authority to perform the specified SQL ALTER statement. If you hold a primary- or secondary-authorization ID that has the proper authority, you can change the Current SQLID to that authorization ID and complete the CREATE. To change the Current SQLID, use the SET command.

B (optional) In the Edit options field, type Y to modify the default values for the options on the Confirm SQL panel. Then, press Enter.

The Options panel is displayed. In the Edit SQL and Confirm options field, type Y to display the options for the Confirm SQL panel. Press END to return to the Confirm SQL panel.

C (optional) In the Edit SQL field, type Y to invoke an ISPF edit session to edit the SQL statement. Then, press Enter.

Press END to save the SQL and return to the Confirm SQL panel.

D (optional) In the Save SQL field, type Y to save the SQL in the CATALOG MANAGER SQL_Table. In the Name of saved SQL field, type a name for the SQL. Then, press Enter.

The saved SQL uses the ID displayed in the Current SQLID field as the object qualifier. If the SQL is not saved, the ID in the Current SQLID is used only to identify DB2 authority.

E (optional) In the Execute field, type Y to execute the SQL displayed on the Confirm SQL panel. Then, press Enter.

The SQL Progress Indicator panel is displayed. The panel automatically refreshes to display the status of the SQL that is being executed.
Extended SQL processing

Use Extended Structured Query Language (ESQL) processing to test Data Manipulation Language (DML) statements with or without host variables.

To test host variables in DML statements

The following procedure describes how to use extended SQL processing to test host variables as you execute a SELECT, DELETE, INSERT, or UPDATE statement.

1 Issue a DML command against a table or view:

   A Generate a list of tables or views. For information, see “Generating lists in CATALOG MANAGER” on page 47.

   B In the Cmd (C) column, type SELECT, DELETE, INSERT, or UPDATE next to a table name.

   C Press Enter.

      Model SQL for the statement is displayed in edit mode.

2 Specify one or more host variables:

   A In the WHERE statement, delete any comment hyphens.

   B Set one or more column values to a host variable name, such as :DATE or :FUNCTION.

   C Press END.

      The Confirm SQL panel is displayed (Figure 90).
3 In the **Execute SQL** field, type **Y**.

4 Press **Enter**.

The SQL Host Variables List panel is displayed (Figure 91).
5 In the Substitute Value column of the DATE row, type the value for the DATE host variable.

6 In the Substitute Value column of the FUNCTION row, type the value for the FUNCTION host variable.

7 Choose one of the following actions:

- To see the SELECT statement that CATALOG MANAGER creates, in the Display SQL field, type Y. Press Enter.

  The SQL Statement Display panel is displayed. You can only view the SQL statement.

- To execute the SELECT statement after specifying the substitute values, in the Execute SQL field, type Y. Press Enter.

  CATALOG MANAGER performs the search.

  If you type N in the Execute SQL field, CATALOG MANAGER displays the SQL Host Variables List panel, from which you can continue to make changes to the substitute values. CATALOG MANAGER does not perform the search until you type Y in the Execute SQL field and press Enter.
Where to go from here

An important feature of CATALOG MANAGER is the ability to view and edit the data in the catalog tables without exiting the product. Chapter 6, “Browsing and editing data,” explains how to invoke the data editing and browsing function after you have created lists of the tables or views with which you want to work.
Browsing and editing data

This chapter contains the following topics:

- **Overview** .......................................................... 223
- **Browsing table data** .............................................. 224
  - Invoking the data browsing function .................. 224
  - Setting options for browsing data .................... 224
  - Browsing data ................................................. 229
  - Browsing data in LOB columns ....................... 230
- **Editing table data** ............................................. 231
  - Invoking the data editing function .................. 231
  - Setting options for editing data ..................... 231
  - Editing data ................................................. 234
- **Copying table data** .......................................... 237
  - Using the COPY command ................................ 237
  - Using the Copy Table Rows option .................. 241
- **Where to go from here** ..................................... 245

**Overview**

By using the data browsing and data editing functions, you can browse, edit, or create data in tables, and in views with the following general characteristics:

- The view is created from a single table.
- The column names in the view are the same as in the table.

You can also browse data contained in large object (LOB) columns and materialized query tables (MQTs).

CATALOG MANAGER enables you to display rows either horizontally or vertically, issue common ISPF commands and new CATALOG MANAGER commands, and use host variables to manage your data easily.
By using the COPY feature of data editing, you can populate newly created tables and views quickly and avoid the need for utilities.

For a list of the commands that you can use with the browsing and data editing features, see Appendix E, “Commands.”

Browsing table data

With CATALOG MANAGER, you can view the data in the catalog tables without exiting the product. You can also connect to a remote DB2 SSID to edit data. For more information, see Chapter 4, “Accessing other DB2 subsystems.”

Invoking the data browsing function

Table 38 lists the available methods for invoking the data browsing function.

Table 38 Commands to invoke data browsing

<table>
<thead>
<tr>
<th>Data browsing command</th>
<th>Where to enter command</th>
</tr>
</thead>
<tbody>
<tr>
<td>BROWSE TB owner.tablename</td>
<td>Command line of Primary Menu panel or list panel</td>
</tr>
<tr>
<td>BROWSE BR</td>
<td>Cmd (C) column of source table on Table List panel</td>
</tr>
<tr>
<td>B</td>
<td>Edit/Browse field on Confirm SQL panel for editing</td>
</tr>
<tr>
<td>TBBR</td>
<td>SELECT SQL statement line of SQL_Table List panel</td>
</tr>
</tbody>
</table>

Setting options for browsing data

1 Invoke the data browsing function.

For more information, see “Invoking the data browsing function” on page 224.

The Browse DB2 Table Options panel is displayed (Figure 92).
**Figure 92  Browse DB2 Table Options panel**

```
DEFF-R ------------------ Browse DB2 Table Options -------------------------
Command ===>
Specify the following options. Press ENTER to continue or END to exit.
Current SQLID            RDACRJ       Current Authid .. RDACRJ
Table name or pattern    QZU.QZUT00_DSC30528
Edit select statement ... N  Display and edit select statement.
Save/Retrieve Select . . . N  (S/R/N) N-no action
                            S-save current select statement in SQL
table using the select statement name
                            R-retrieve list of saved select statements
                            matching pattern in select statement name
Select statement name ...
Display selected row count N  Display count of rows matching WHERE clause.
Select row limit ........ 300  0-99999999 rows to edit. 0 = no limit.
Initial Display Mode .... C  C-column view, R-row view
Display additional options N  Display panel to specify browsing options.
```

2 In the **Current SQLID** field, type a different SQLID.

3 In the **Table name or pattern** field, type the name of a table or type a pattern.

---

**NOTE**

Wildcards are valid in the **Table name or pattern** field. For information on wildcards, see “Including wildcards in qualifiers” on page 42.

---

4 In the **Edit select statement** field, type Y to display or edit the SELECT statement that CATALOG MANAGER generates.

The Select Statement Specification panel is displayed (Figure 93).
You can customize the SELECT statement by making the specifications that are shown in Table 39 on the Select Statement Specification panel. By default, all columns are designated as “selected.”

Table 39  SELECT statement specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>remove a column from the SELECT statement(^a)</td>
<td>delete the S designator beside the column name</td>
</tr>
<tr>
<td>specify a sequence of columns(^a)</td>
<td>replace the S designator with a number from 0 through 99</td>
</tr>
<tr>
<td>specify a sort order</td>
<td>type A (ascending) or D (descending) for the value in the Order field</td>
</tr>
<tr>
<td>specify the type of comparison</td>
<td>type a valid operator in the Oper field (see Table 34 on page 186)</td>
</tr>
<tr>
<td>specify values for the search operation(^b)</td>
<td>type the values in the Value field</td>
</tr>
</tbody>
</table>

\(^a\) You can use the S designator for some columns and a sequence number for other columns of the same table.

\(^b\) If you specify a search value for a column in the Value field, then the applicable column will be included in the SELECT statement even if you do not select the column with the S designator or a sequence number.
After customizing the SELECT statement, press **Enter** to display the panel with your specifications.

*(optional)* Use the command-line commands that are shown in Table 40 to clear changes that you have made.

**Table 40 SELECT statement specification panel commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESET</td>
<td>clears S designators and sequence numbers</td>
</tr>
<tr>
<td>RESET ALL</td>
<td>clears S designators, sequence numbers, and user input in the Order, Oper(ator), and Value fields</td>
</tr>
<tr>
<td>RESTART</td>
<td>ignores user changes and reinitializes the panel</td>
</tr>
</tbody>
</table>

Press **END** to display the Browse DB2 Table Options panel.

**5** In the **Save/Retrieve Select** field, type **S**, **R**, or **N** to indicate your action on the SELECT statement:

<table>
<thead>
<tr>
<th>To do this</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>save the SELECT statement in the SQL_Table</td>
<td>S</td>
</tr>
<tr>
<td>retrieve a list of SELECT statements from the SQL_Table</td>
<td>R</td>
</tr>
<tr>
<td>CATALOG MANAGER attempts to retrieve a SELECT statement that matches a member name or pattern in the <strong>Select statement name</strong> field.</td>
<td></td>
</tr>
<tr>
<td>discard the SELECT statement and not retrieve a SELECT statement from the SQL_Table</td>
<td>N</td>
</tr>
</tbody>
</table>

**6** If you typed **R** in the **Save/Retrieve Select** field in **step 5**, in the **Select statement name** field, type a member name or pattern. CATALOG MANAGER attempts to match an entry in the SQL_Table with the member name or pattern that you type in the **Select statement name** field.

**7** In the **Display selected row count** field, type **Y** or **N** to display the number of rows that contain data that satisfies the WHERE clause.

**8** In the **Select row limit** field, type a value to indicate the number of rows that are displayed.

---

**NOTE**

The value in the **Select row limit** field overrides the value that is set in the **Max Lines per list** field on the Options panel, but only for the current editing or browsing session.
In the **Initial Display Mode** field, type C or R to indicate the display mode:

- **C** (column view) displays column names and data horizontally. CATALOG MANAGER formats each row as a separate line on the display.

- **R** (row view) displays column names and data vertically. CATALOG MANAGER wraps the larger data columns within the panel, enabling you to view without scrolling left and right.

Changes to the default view are saved in the user’s profile.

In the **Display additional options** field, type Y or N to display the Edit and Browse Options panel (Figure 94), from which you can specify additional values.

### Figure 94  Edit and Browse Options panel

```
DBDC-R -------------------  Edit and Browse Options  -------------------------
Command ==>
More:     +
Specify the following options, then press END to exit.
Display Headings . . . . . 1  1-Names, 2-Labels, 3-Numbers
Date and Time defaults . . 2  Select value for columns with no DB2 default
  1 - 1900-01-01-00.00.00.000000
  2 - Current date and time
  3 - Blank
Display update SQL . . . . N  Display SQL used to update the table.
Confirm before update . . . N  Display confirm panel before saving changes.
Display Statistics . . . . Y  Display Table Editing Statistics.
Browse with UR . . . . . . . Y  Browse DB2 rows using WITH UR in select.
CAPS ON . . . . . . . . . . N  Uppercase input and modified values.
Default SQL Owner . . . . . S  Default owner for SQL unqualified sql table
  member names when saving or retrieving SQL.
  T - TSO ID
  S - SQL ID
Memory Allocation Limit .. 0  Meg. Maximum memory to allocate for
  DB2 rows, in megabytes. 0 = No limit.
Left Justify Numerics . . . N  Numeric fields in rowview mode will
  be displayed left justified.
Display Decimal Point . . . N  Include a decimal point in the value
  of columns with type DECIMAL(n,0)
Clear Editor Users Table .. N  (Y/N) A SYSADM may need to delete ALL rows
  from the editor users table, where Catalog
  Manager records who is editing which tables.
```
Table 41 shows the options that you can specify on the Edit and Browse Options panel.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Headings</td>
<td>selects the style of column heading</td>
</tr>
<tr>
<td>Date and Time defaults</td>
<td><em>(editing data)</em> specifies whether the date, time, and timestamp columns that are initialized when a new row is inserted should use the time when the new row is created or the time when the row is inserted into the DB2 table</td>
</tr>
<tr>
<td>Display update SQL</td>
<td><em>(editing data)</em> specifies whether to display (after INSERT, UPDATE, and DELETE statements are executed) SQL statements that were executed while saving your changes</td>
</tr>
<tr>
<td>Confirm before update</td>
<td><em>(editing data)</em> specifies whether to display a confirmation message to save your changes or to commit the changes</td>
</tr>
<tr>
<td>Display Statistics</td>
<td>*(editing data) after editing a table or view, displays the number of INSERTs, FETCHes, UPDATEs, and DELETEs that have been performed</td>
</tr>
<tr>
<td>Browse with UR</td>
<td><em>(browsing data)</em> specifies whether to append a WITH UR (with uncommitted read) clause to the SELECT statement The WITH UR clause is not applicable when the SELECT statement is imported or is provided from the CATALOG MANAGER SQL_Table.</td>
</tr>
<tr>
<td>CAPS ON</td>
<td><em>(editing data)</em> specifies whether all applicable data that is typed during the editing session should be uppercase This option value is also saved in the user’s profile.</td>
</tr>
<tr>
<td>Default SQL Owner</td>
<td>specifies whether SQL ID or TSO ID is the default owner when the table or view is saved to or retrieved from the SQL_Table</td>
</tr>
<tr>
<td>Memory Allocation Limit</td>
<td>specifies the maximum amount of memory (as number of megabytes) that CATALOG MANAGER should allocate to hold rows that are fetched from DB2</td>
</tr>
<tr>
<td>Left Justify Numerics</td>
<td><em>(row view)</em> specifies whether to left justify numeric fields</td>
</tr>
<tr>
<td>Display Decimal Point</td>
<td>specifies whether to display the decimal point character in columns defined as DECIMAL(n,0)</td>
</tr>
<tr>
<td>Clear Editor Users Table</td>
<td><em>(editing data)</em> deletes all rows from the BMCACTvr.Ver_EDITOR_USERS table (see “CATALOG MANAGER tables” on page 156) This option is available only to users with SYSADM authority.</td>
</tr>
</tbody>
</table>

Browsing data

1 Invoke the data browsing function. For more information, see “Invoking the data browsing function” on page 224.

2 Set your options for browsing data. For more information, see “Setting options for browsing data” on page 224.
3 Press Enter.

The Browse DB2 Table panel is displayed.

4 After you have browsed the data, press END.

**Browsing data in LOB columns**

Use the following procedure to browse data in a LOB column.

---

**NOTE**

You cannot use the data editing function to edit data in tables that contain LOB columns.

---

**Before you begin**

CATALOG MANAGER requires a TSO region size of 8 MB or greater. If you have a character large object (CLOB) column that contains up to 2 MB of data, increase the TSO region size.

**To browse LOB data**

1 Generate a table list.

2 Invoke the data browsing function.

   For more information, see “Invoking the data browsing function” on page 224.

3 In the Browse DB2 Table Options panel, specify your options.

   For more information, see “Setting options for browsing data” on page 224.

   The Browse DB2 Table panel is displayed.

4 To view the entire value for a LOB column, perform the following steps:

   A On the **Command** line, type **ZOOM (Z)**.

   B Position your cursor on the value that you want to view.
Press Enter.

The data is displayed in an ISPF Browse panel.

Press END to return to the Browse DB2 Table panel.

## Editing table data

CATALOG MANAGER provides you with the ability to edit the data in the catalog tables without exiting the product.

### Invoking the data editing function

Table 42 lists the available methods for invoking the data editing function.

<table>
<thead>
<tr>
<th>Data editing command</th>
<th>Where to enter command</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDIT TB <code>owner.tablename</code></td>
<td>Command line of Primary Menu panel or list panel</td>
</tr>
<tr>
<td>EDIT ED</td>
<td>Cmd (C) column of source table on Table List panel</td>
</tr>
<tr>
<td>E</td>
<td>Edit/Browse field on Confirm SQL panel for editing SELECT SQL statement</td>
</tr>
<tr>
<td>TBEDIT</td>
<td>SELECT SQL statement line of SQL_Table List panel</td>
</tr>
</tbody>
</table>

### Setting options for editing data

1. Invoke the data editing function. For more information, see “Invoking the data editing function” on page 231.

The Edit DB2 Table Options panel is displayed (Figure 95). The Edit DB2 Table Options panel provides the same options as the Browse DB2 Table Options panel (Figure 92), with the following additions:

- Hold rows during edit
- Edit or Browse mode
- Copy Table Rows
2 Follow the steps in “Setting options for browsing data” on page 224.

3 In the **Hold rows during edit** field, type **T**, **R**, or **N** to indicate how requests for edits from other users are handled while you are editing data:

<table>
<thead>
<tr>
<th>To specify this locking option</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>shared table lock</td>
<td>T</td>
</tr>
</tbody>
</table>

CATALOG MANAGER issues the SQL LOCK TABLE `owner.tablename` MODE SHARE statement.

Other users can browse rows in the table that you are editing, but they cannot edit any data in the table until your changes have been applied and committed.

Any changes that you make are committed when you exit the editing function to the Edit DB2 Table Options panel.
### Setting options for editing data

**Chapter 6: Browsing and editing data**

Availability of these locking options is determined during the installation of CATALOG MANAGER by the ELO (Editor Lock Options) setting in the BMCDB2 and BMCADMF2 CLISTs. Check with your system administrator or CATALOG MANAGER installer if the default locking options are incorrect for your tasks. For more information about locks, see “Handling lock contention” on page 234.

<table>
<thead>
<tr>
<th><strong>To specify this locking option</strong></th>
<th><strong>Type</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>row lock</td>
<td>R</td>
</tr>
<tr>
<td>CATALOG MANAGER does not issue any locks and allows DB2 to perform normal lock escalation. If the table space in which you are editing was created with LOCKSIZE ROW, row locks can be used unless DB2 performs lock escalation based on the number of rows edited. Any changes that you make are committed when you exit the editing function to the Edit DB2 Table Options panel.</td>
<td></td>
</tr>
<tr>
<td>no lock</td>
<td>N</td>
</tr>
<tr>
<td>CATALOG MANAGER does not issue any locks and allows DB2 to perform normal lock escalation. After data has been retrieved, a COMMIT is performed that releases all locks. After edits have been completed, CATALOG MANAGER compares the data that was originally read with the current data in the table.</td>
<td></td>
</tr>
</tbody>
</table>

- If no changes have occurred, the edits are applied and committed.
- If the data in the table has changed from the data that was originally read, CATALOG MANAGER asks whether your edits should overwrite those found in the table currently. If you respond Yes, the edits are applied and committed.

To enable Fast Path Navigation when locking options for data editing have been specified, the installer must enable the locking options command in the BMCADMF2 CLIST.

4. In the **Edit or Browse mode** field, type E to edit data or B to browse data.

Because data editing requires a higher authorization level than data browsing, the **Edit or Browse mode** option is not available to users who have invoked the data browsing function under the following conditions:

- The Browse DB2 Table Options panel is specified as an initial entry panel for CATALOG MANAGER.
- Data browsing was originally invoked by the issuance of one of the data browsing commands (Table 42).
In the **Copy Table Rows** field, type **Y** or **N** to copy rows from a source table or view to a target table or view. CATALOG MANAGER formats SELECT and INSERT statements based on values that you specify on a sequence of panels.

### Handling lock contention

When CATALOG MANAGER fetches rows from a table for editing, the rows are read into memory. DB2 performs normal locking as part of reading the rows. For example, if the table space contains only one table and the SELECT statement has no **WHERE** clause, DB2 might escalate row or page locks to a table space lock, since all rows in the table are being read.

If you are browsing data, the SELECT statement is modified to append a **FOR FETCH ONLY** clause to prevent DB2 from holding locks for fetched rows. Data browsing performs as if the **Hold rows during edit** option was set to **N**. Because data cannot be edited in data browsing mode, comparing and overwriting data are meaningless.

How these locks affect your other applications depends on what isolation level you set when you bound packages for CATALOG MANAGER and other applications, and what lock options were specified when the table space containing the table was created.

For more information about how locking affects your applications, see the installation guide and the IBM documentation.

### Editing data

1. Invoke the data editing function. For more information, see “Invoking the data editing function” on page 231.

2. Set the options for controlling the display and the SQL processing. For more information, see “Setting options for editing data” on page 231.

3. Press Enter.

   If another user has begun an editing session in the table or view, the **Other Users Editing This Table** panel is displayed. To edit the table or view, press Enter. To display the **Edit DB2 Table Options** panel, press END or CANCEL.
If no other users are editing the table, the Edit DB2 Table panel is displayed (Figure 96) according to the Initial Display Mode value that was selected in the Edit DB2 Table Options panel (Figure 95).

Figure 96  Edit DB2 Table panel in column view mode

4 Edit the data by moving to the column and row that you want to change and typing the new values. You can switch between viewing modes as follows:

- To switch from column view to row view, type the ROW command on the Command line, and then press Enter.

- To switch back to column view, press END.

Any changes that you have made to the data are highlighted before the viewing mode is switched.

5 If the address space is insufficient to display all of the selected rows, CATALOG MANAGER displays a message that refers to memory constraints. To display more rows, complete the following steps:

A  Ensure that you are in column view mode.

B  On the Command line, type MORE.
C Press Enter.

The Display More Rows Options panel is displayed (Figure 97).

Figure 97  Display More Rows Options panel

D In the Save table changes field, type Y to save the edits that you have made since you last saved the changes.

E To display the next set of rows in the table, press Enter.

The Edit DB2 Table panel is displayed with the next set of rows.

NOTE
If the message that refers to memory constraints is displayed again, repeat step 5 until all of the selected rows have been displayed and edited.

6 Press SAVE or END to terminate the editing session.

7 (optional) If you typed Y for the Confirm before update value on the Edit and Browse Options panel, the Confirm Updates popup is displayed.
   - Press Enter to save the changes and end the editing session.
   - Press END to cancel saving the changes and return to the editing session.
8 (optional) If you typed Y for the Display Statistics value on the Edit and Browse Options panel, the Statistics popup is displayed. You can commit the edits or rollback the data to its original value.

When the editing session is ended, the Edit DB2 Table Options panel is displayed. The panel includes a confirmation message.

Copying table data

With CATALOG MANAGER, you can easily copy data from one table or view into another table or view. The copy action functions within a range of sequential columns that begins with the first column in both the source and target tables or views.

To perform a copy action, the data formats in corresponding columns of each table must be compatible, as follows:

- Any numeric data type is compatible with any other numeric data type.
- CHAR, VARCHAR, and LONG VARCHAR data types are compatible.
- All other data formats are compatible if the data type and length are identical.

Using the COPY command

This procedure describes how to use the COPY command during a data editing session. This method is efficient in the following situations:

- You need to make several changes to the initial SELECT statement.
- You need to update the retrieved rows before committing the copy action.

1 Invoke a data editing session for the target table or view. For information, see “Invoking the data editing function” on page 231.

2 On the Edit DB2 Table Options panel, set option values for editing data. For information, see “Setting options for editing data” on page 231.

**NOTE**

To copy data by using the COPY command, the following option values are required:

- **Initial Display Mode**=C (column view)
- **Edit or Browse Mode**=E (edit data)
3 Press Enter.

- If another user has begun an editing session in the table or view, the Other Users Editing This Table panel is displayed. To edit the table or view, press Enter. To display the Edit DB2 Table Options panel, press END or CANCEL.

- If no other users are editing the table, the Edit DB2 Table panel is displayed.

4 On the Command line, type COPY.

5 Press Enter.

The COPY From DB2 Table Options panel is displayed (Figure 98).

**Figure 98  COPY From DB2 Table Options panel**

```
DEFF-R ----------------- COPY From DB2 Table Options -----------------
Command ===> Specified the following options, then press ENTER to read table rows.
Press END or CANCEL to abandon the copy.
Copy source table . . . . . . QZU.QZUT01_DCII9S05
Edit COPY select statement . . Y Display and edit copy select statement.
Save/Retrieve Select . . . . N (S/R/N) N-no action
    S-save current select statement in SQL
    table using the select statement name
    R-retrieve list of saved select statements
    matching pattern in select statement name
Select statement name . . .
Number of rows to Copy . . . 0  0-99999999 rows to copy. 0 = no limit.
Approximate upper limit is shown.
```

6 Set options for specifying the source table or view and for customizing the SELECT statement that CATALOG MANAGER creates to perform the copy.

To specify the source table or view, you can use the following methods:

- Type the name of a table or view.

- Type a pattern that includes a wildcard. For more information about using wildcards, see “Including wildcards in qualifiers” on page 42.
Retrieve a saved SELECT statement from the SQL_Table by typing Y in the Save/Retrieve Select field. This option also enables you to save the SELECT statement that CATALOG MANAGER creates for the current copy action.

In this example, the following options have been selected:

- to copy from the source table
- to edit the SELECT statement
- to copy all applicable rows

7 Press Enter.

The Select Statement Specification panel is displayed (Figure 99).

**Figure 99  Select Statement Specification panel**

You can substitute host variables for the Value specifications in the SELECT statement. For more information, see “Using host variables in a search” on page 190.
9 Press END.

The COPY From DB2 Table Options panel is displayed. You can specify whether to change other options on the panel.

10 When you have determined the options to use for the SELECT statement, press Enter.

The Edit DB2 Table panel is displayed to show you the rows that have been copied (Figure 100).

**Figure 100 Edit DB2 Table panel showing rows that were copied**

```
DEFF --------------------------  Edit DB2 Table  -----------------------------
Command ===>                                                  Scroll ===> CSR
QZU.QZUTO1_DCII9505 (301/2288)
****** COLUMN_1 COLUMN_2 COLUMN_3 COLUMN_4 COLUMN_5 COLUMN_6
*INS 1818599  0  AABN0000467 1124064  4064  'DEFENDANT ARRAIGN
*INS 1818629  7364  AABN0000478 1130064  0064  'DEFENDANT ARRAIGN
*INS 1818687  26700  AABN0000495 1139064  9064  'DEFENDANT ARRAIGN
*INS 1818706 -20104  AABN0000501 1141064  1064  'DEFENDANT ARRAIGN
*INS 1818722  27713  AABN0000506 1145064  5064  'DEFENDANT ARRAIGN
*INS 1818733 -14913  AABN0000510 1146063  6063  'DEFENDANT ARRAIGN
*INS 1818754  7200  AABN0000517 1148064  8064  'DEFENDANT ARRAIGN
*INS 1818758 16000  AABN0000519 1148064  8064  'DEFENDANT ARRAIGN
*INS 1818781  7300  AABN0000525 1160064  0064  'DEFENDANT ARRAIGN
*INS 1818810  3300  AABN0000536 1164064  4064  'DEFENDANT ARRAIGN
*INS 1818836  6800  AABN0000543 1171064  1064  'DEFENDANT ARRAIGN
*INS 1818895  0  AABN0000561 1176064  6064  'DEFENDANT ARRAIGN
*INS 1818902  0  AABN0000562 1177064  7064  'DEFENDANT ARRAIGN
*INS 1818932  4275  AABN0000573 1182064  2064  'DEFENDANT ARRAIGN
*INS 1818944  0  AABN0000577 1184064  4064  'DEFENDANT ARRAIGN
*INS 1818954  5985  AABN0000578 1185064  5064  'DEFENDANT ARRAIGN
*INS 1447973 18000  AABN0000585 1655233  5233  'DEFENDANT ARRAIGN
*INS 1819036  0  AABN0000600 1209064  9064  'DEFENDANT ARRAIGN
*INS 1448522  0  AABN0000603 0773625  3625  'DEFENDANT ARRAIGN
*INS 1448585 -31536  AABN0000612 0774002  4002  'DEFENDANT ARRAIGN
```

**NOTE**
The changes are committed unless you have specified Y for the Confirm before update option on the Edit and Browse Options panel.

11 Press END to display the Edit DB2 Table Options panel. CATALOG MANAGER displays the CHANGES COMMITTED message to confirm the COPY action.
Using the Copy Table Rows option

This procedure describes how to copy data by specifying an option on the Edit DB2 Table Options panel. This method is efficient for copying data from a large number of rows when you require few changes to the SELECT or INSERT statements that CATALOG MANAGER creates to perform the copy action.

1. Invoke a data editing session for the *target* table or view. For information, see “Invoking the data editing function” on page 231.

2. On the Edit DB2 Table Options panel, set option values for editing data. For information, see “Setting options for editing data” on page 231.

---

**NOTE**

To copy data by using the Copy Table Rows option, the following option values are required:

- **Edit or Browse Mode** = E (edit data)
- **Copy Table Rows** = Y

---

3. Press Enter.

   If another user has begun an editing session in the table or view, the Other Users Editing This Table panel is displayed. To edit the table or view, press Enter. To display the Edit DB2 Table Options panel, press END or CANCEL.

   If no other users are editing the table, the Copy Table Rows Specifications panel is displayed.

4. Set options for specifying the source and target tables or views and for customizing the SELECT and INSERT statements that CATALOG MANAGER creates to perform the copy (Figure 101).
Using the Copy Table Rows option

To specify the source table, you can type the name of a table or view or use a pattern that includes a wildcard. For more information about using wildcards, see “Including wildcards in qualifiers” on page 42.

In this example, the following options have been selected:

- to copy from a table
- to edit the SELECT statement
- to edit the INSERT statement

5 Press Enter.

6 When you have specified all of the options, press Enter.

The Select Statement Specification panel is displayed.

7 Customize the SELECT statement by modifying the selected columns and values on the panel (Figure 102). Press HELP to display example specifications.
Figure 102  Select Statement Specification panel

<table>
<thead>
<tr>
<th>S</th>
<th>NAME</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>ORDER</th>
<th>OPER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>COLUMN_1</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_2</td>
<td>SMALLINT</td>
<td>2</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_3</td>
<td>CHAR</td>
<td>12</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_4</td>
<td>CHAR</td>
<td>7</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_5</td>
<td>CHAR</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_6</td>
<td>VARCHAR</td>
<td>55</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_7</td>
<td>DECIMAL</td>
<td>31,20</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_8</td>
<td>DECIMAL</td>
<td>11,2</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_9</td>
<td>SMALLINT</td>
<td>2</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_10</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_11</td>
<td>FLOAT</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_12</td>
<td>FLOAT</td>
<td>8</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_13</td>
<td>DATE</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_14</td>
<td>TIME</td>
<td>3</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_15</td>
<td>TIMESTMP</td>
<td>10</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_16</td>
<td>VARCHAR</td>
<td>30</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_17</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_18</td>
<td>CHAR</td>
<td>20</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_19</td>
<td>CHAR</td>
<td>24</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>COLUMN_20</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE**

You can substitute host variables for the Value specifications in the SELECT statement. For more information, see “Using host variables in a search” on page 190.

8 Press END.

The Copy Table Rows Specifications panel is displayed. You can specify whether to change other options on the panel.

9 Press Enter.

An ISPF edit panel is displayed. You can edit the INSERT SQL statement to ensure that the column sequence is compatible with the column sequence in the SELECT statement (Figure 103).
Figure 103  ISPF panel for editing INSERT statement

10  Save the changes to the INSERT statement and close the ISPF edit panel.

The Copy Table Rows Specifications panel is displayed.

11  Choose one of the following actions:

- To complete the copy action, press Enter.

  The Edit DB2 Table Options panel is displayed with the CHANGES COMMITTED message.

- To cancel the copy action, press END or CANCEL.

  The Edit DB2 Table Options panel is displayed with the COPY CANCELLED message.
Creating lists of objects that exist in your DB2 catalog and issuing commands against those objects to get the data that you need are the core functions of the CATALOG MANAGER product. Many more features can facilitate how you interact with the DB2 catalog.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create objects</td>
<td>Chapter 7, “Creating objects”</td>
</tr>
<tr>
<td>Drop and recover objects</td>
<td>Chapter 8, “Dropping and recovering objects”</td>
</tr>
</tbody>
</table>
Creating objects

This chapter contains the following topics:

Overview ................................................................. 247
Before you begin ...................................................... 247
Estimating space requirements based on user-specified values .......... 248
  Estimating space requirements for a table space 249
  Estimating space requirements for an index 252
Using an existing object as a model to create objects 253
  Considerations for creating objects 253
  Example of creating a table 254
Generating DDL to create objects 267
Where to go from here 270

Overview

Creating objects is one of the more challenging tasks associated with maintaining a database. The methods traditionally used to create DB2 database objects are time-consuming, and they require an extensive knowledge of both Structured Query Language (SQL) and the existing DB2 catalog table structures.

CATALOG MANAGER provides simpler, more efficient methods for handling database definition by enabling you to create objects if you do not know SQL. You can create objects by using an existing object as a model. In addition, you can generate DDL to create objects.

Before you begin

Before you begin creating objects, you should be familiar with how to search for and list objects. If necessary, review Chapter 5, “Working with lists and searches” before reading this chapter.
Estimating space requirements based on user-specified values

By default, CATALOG MANAGER specifies your SQL ID as the owner when you create objects. To specify that your TSO ID is the owner, set the value of the SQL Ownerid switch to TSO. For more information, see “Setting CATALOG MANAGER switches” on page 95.

Before you create table spaces or indexes, you can use the Simple Space Estimation (SSE) feature to estimate the space requirements of these objects and predict storage capacity. For information, see “Estimating space requirements based on user-specified values” on page 248.

NOTE
You must have the appropriate DB2 authorization to create objects before you can use CATALOG MANAGER to perform the tasks described in this chapter.

Before you create table spaces or indexes, you can use the Simple Space Estimation (SSE) feature to estimate the space requirements of these objects and predict storage capacity. For information, see “Estimating space requirements based on user-specified values” on page 248.

NOTE
You do not need to run BMCSTATS before using SSE.

For example, for a non-partitioned table space estimate, you can change the fields highlighted in Figure 104. SSE then estimates how much space will be required if you reorganize the table space. SSE displays estimates for the entire table space in the Estimated box on the right, and table-level estimates at the bottom of the panel.

For a description of each field that accepts user-specified values, see the online Help.
Estimating space requirements for a table space

You can use SSE to estimate space requirements for non-partitioned, partitioned, and partition-by-growth table spaces:

- “To estimate space requirements for a non-partitioned table space” on page 249
- “To estimate space requirements for a partitioned table space” on page 250
- “To estimate space requirements for a partition-by-growth table space” on page 251

**NOTE**

Space estimation is not available for XML objects or for objects that have partial statistics.

To estimate space requirements for a non-partitioned table space

1. From the command prompt, type `SSE` and press **Enter**.
2. On the Space Estimation panel, enter `TS` in the **Object Type** field and press **Enter**.
3 Estimate space requirements for the entire table space or for individual tables as follows:

A Change any of the modifiable values (highlighted in Figure 104 on page 249) as needed to correspond to your table space.

NOTE
For more information about a specific field, press F1.

B Press Enter to update the Estimated fields:

- The Estimated box on the right side of the panel displays estimates for the entire table space.
- The Estimated list in the bottom right corner shows estimates per table.

NOTE
The bottom of the panel shows more information about the table. In the Cmd column, you can enter I to insert, R to repeat, or D to delete lines.

4 (optional) If you want to see additional estimates based on different values, repeat step 3.

5 When finished, press END to exit.

To estimate space requirements for a partitioned table space

1 From the command prompt, type SSE and press Enter.

2 On the Space Estimation panel, enter TS in the Object Type field, and P in the Tablespace Type field. Then, press Enter.

3 Estimate space requirements for the entire table space or for individual partitions as follows:

A Change any of the modifiable values (highlighted in Figure 104 on page 249) as needed to correspond to your table space.

NOTE
For more information about a specific field, press F1.
B Press Enter to update the Estimated fields:

- The Estimated box on the right side of the panel displays estimates for the entire table space.
- The Estimated list in the bottom right corner shows estimates per partition.

**NOTE**
The bottom of the panel shows more information about the table. In the Cmd column, you can enter I to insert, R to repeat, or D to delete lines.

4 (optional) If you want to see additional estimates based on different values, repeat step 3.

**NOTE**
This space estimation function does not support the Average Length feature for table columns.

5 When finished, press END to exit.

**To estimate space requirements for a partition-by-growth table space**

1 From the command prompt, type SSE and press Enter.

2 On the Space Estimation panel, enter TS in the Object Type field, and G in the Tablespace Type field. Then, press Enter.

3 Estimate space requirements for the entire table space or for individual partitions are as follows:

A Change any of the modifiable values (highlighted in Figure 104 on page 249) as needed to correspond to your table space.

**NOTE**
For more information about a specific field, press F1.

B Press Enter to update the Estimated fields:

- The Estimated box on the right side of the panel displays estimates for the entire table space.
- The bottom of the panel shows more information about the table itself.
4 (optional) If you want to see additional estimates based on different values, repeat step 3.

5 When finished, press END to exit.

Estimating space requirements for an index

You can use SSE to estimate space requirements for an index.

NOTE
Space estimation is not available for XML objects or for objects that have partial statistics.

To estimate space requirements for an index

1 From the command prompt, type SSE and press Enter.

2 On the Space Estimation panel, enter IX in the Object Type field, and press Enter.

3 Estimate space requirements for the index as follows:

   A Change any of the modifiable values (highlighted in Figure 104 on page 249) as needed to correspond to your index.

   NOTE
   For more information about a specific field, press F1.

   B Press Enter to update the Estimated fields:

   ■ The Estimated box on the right side of the panel displays estimates for the index.

   ■ The Estimated list in the bottom right corner shows estimates for the index.

   NOTE
   The bottom of the panel shows more information about the index. In the Cmd column, you can enter I to insert, R to repeat, or D to delete lines.
4 (optional) If you want to see additional estimates based on different values, repeat step 3.

5 When finished, press END to exit.

**Using an existing object as a model to create objects**

You can use the design of an existing DB2 object as a *model* to create a new object. CATALOG MANAGER makes it easy for you to change only the attributes that should differ from the model. The process for creating objects is as follows:

1. Define the object.
   A. Create a list.
   B. Type `CREATE (CR)` next to the object.

2. Specify the attributes for the object.

3. Generate SQL.

**Considerations for creating objects**

Consider the following items when you create objects:

- With CATALOG MANAGER, you can create table spaces explicitly or implicitly, as you can in DB2.
- When you generate a list, you can specify an object qualifier in the **Qualifier** field of the CATALOG MANAGER Primary Menu panel. For more information, see “Specifying a qualifier” on page 42.
- To obtain a list of valid values for a field, type `?` in the input field.
- To display the full value of an object with a long name, press ZOOM (F4) in the input field.
- To enter a name that is longer than 18 characters, press ZOOM (F4).
Example of creating a table

For more information about the fields on the panel, press HELP when the cursor is on a field.

To create an object that is identical in structure to the existing object, accept all of the default attribute values.

Example of creating a table

This example demonstrates how to use CATALOG MANAGER on a DB2 Version 10 subsystem to create a table from an existing table model.

To define the table

1. Create a table list that includes the table that you want to use as a model. For information, see “Generating lists in CATALOG MANAGER” on page 47.

2. In the Cmd column of the table list, type CREATE (CR) beside the table that you want to use as a model.

3. Press Enter.

The Create/Alter Table panel is displayed (Figure 105). The displayed attribute values match those of the table that you are using as a model.
4 *(optional)* In the **Table owner** field, type a name for the table owner.

5 In the **Table Name** field, type a name for the table that you are creating.

   The name must be unique within the SQL ID of the table owner.

6 Accept or modify the default attribute values shown on the rest of the panel.

**To define additional attributes**

1 In the **Edit Additional options** field, type **Y** to define additional table attributes.

2 Press **Enter**.

   A second Create/Alter Table panel is displayed *(Figure 106)*.
3 Accept or modify the default attribute values shown on the rest of the panel.

4 Press END to return to the first Create/Alter Table panel.

To define the table columns

1 In the Edit column data field, type Y to modify the column definitions for the new table.

2 Press Enter.

The Columns panel is displayed (Figure 107). You can use the ISPF INSERT, DELETE and REPEAT commands to increase or reduce the number of columns.
Specify new values for the column attributes as follows:

A. To change a column name, type a new name in the Name field.

B. To change the schema name to the distinct table type that you are using, edit the value in the Schema field. Leave this value blank if you are using a base table type.

C. To change the data type for a column, edit the value in the Data Type field.

D. To change the length of the field, edit the value in the Length field.

E. To change the number of digits stored to the right of the decimal point, edit the value in the Scale field.

F. To indicate whether null values are allowed in the column, specify Y or N in the Nl field.

G. To indicate whether a default value is placed in a column, specify a value in the Df field.

4. To edit additional column options, in the Edit Opt field, type Y and press Enter.

The Column Options panel is displayed (Figure 108).
To edit additional column options, type **Y** in the **Edit Addtnl column opts** field, and press Enter.

The Additional column options panel is displayed (Figure 109).
A Accept or modify the default attribute values shown on the rest of the panel.

B To edit the comment and label for the column, type Y in the Edit comment and label field, and press Enter.

The Comment and Label panel is displayed.

1. In the Label field, type the text that you want to store as a label for the table. The label text can include up to 30 characters.

2. In the Comment field, type the text that you want to store as a comment for the table. The comment text can include up to 762 characters.

3. Press END to return to the Additional column options panel.

C To edit identity column settings, type Y in the Edit Column Identity field, and press Enter.

**NOTE**

Only a column with a numeric data type can be an identity column, and each table can have only one identity column.

The Column Identity Information panel is displayed (Figure 110).
Example of creating a table

Figure 110 Column Identity Information panel

<table>
<thead>
<tr>
<th>Column Identity Information</th>
<th>1 to 12 of 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>Scroll ===&gt; PAGE</td>
</tr>
<tr>
<td>Column name. . . . . . . . .</td>
<td>COLUMN_1</td>
</tr>
<tr>
<td>Generated. . . . . . . . . .</td>
<td>Always. By default</td>
</tr>
<tr>
<td>Start/Restart with . . . .</td>
<td></td>
</tr>
<tr>
<td>Increment by . . . . . . .</td>
<td>1</td>
</tr>
<tr>
<td>MinValue . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>MaxValue . . . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>Cache. . . . . . . . . . . .</td>
<td>Y</td>
</tr>
<tr>
<td>Cache Amount. . . . . . . .</td>
<td>20</td>
</tr>
<tr>
<td>Cycle. . . . . . . . . . . .</td>
<td>N</td>
</tr>
<tr>
<td>Order. . . . . . . . . . . .</td>
<td>N</td>
</tr>
<tr>
<td>Y/N Preallocate and keep in memory</td>
<td></td>
</tr>
<tr>
<td>Number to preallocate</td>
<td></td>
</tr>
<tr>
<td>Y/N Continue after reaching min/max</td>
<td></td>
</tr>
<tr>
<td>Y/N Generate in order of request</td>
<td></td>
</tr>
</tbody>
</table>

D  Press END three times to return to the Create/Alter Table panel.

To create and edit table constraints

1  In the Edit table constraints field, type Y to create or edit the table constraints.

2  Press Enter.

The Table Constraints panel is displayed (Figure 111).

Figure 111 Table Constraints panel

<table>
<thead>
<tr>
<th>Table Constraints</th>
<th>1 to 6 of 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>Scroll ===&gt; PAGE</td>
</tr>
<tr>
<td>Table Creator. . .</td>
<td>QZU</td>
</tr>
<tr>
<td>Table Name . . . .</td>
<td>QZUT00_DSC30S2B</td>
</tr>
<tr>
<td>Edit Unique/Primary . . .</td>
<td>N</td>
</tr>
<tr>
<td>Edit Foreign Key . . .</td>
<td>N</td>
</tr>
<tr>
<td>Edit Check . . . .</td>
<td>N</td>
</tr>
<tr>
<td>Y/N Edit Unique/Primary Constraints</td>
<td></td>
</tr>
<tr>
<td>Y/N Edit Foreign Key Constraints</td>
<td></td>
</tr>
<tr>
<td>Y/N Edit Check Constraints</td>
<td></td>
</tr>
</tbody>
</table>
3 In the **Edit Unique/Primary** field, type Y to edit the unique and primary constraints. Press Enter.

The Table Unique and Primary Constraints panel is displayed. If the model table contains unique or primary constraints, they are listed on this panel.

A In the **Cmd** column, type I (insert), R (repeat) or D (delete). Press HELP for a description of the fields on the panel.

B Press END to display the Table Constraints panel.

4 In the **Edit Foreign Key** field, type Y to edit foreign keys. Press Enter.

The Table Foreign Key Constraints panel is displayed. If the model table contains foreign key constraints, they are listed on this panel.

---

**NOTE**

To create or drop a foreign key, you must have the ALTER privilege on the parent and dependent tables.

---

A In the **Cmd** column, type I (insert), R (repeat) or D (delete). Press HELP for a description of the fields on the panel.

B Press END to display the Table Constraints panel.

5 In the **Edit Check** field, type Y to edit the check constraints. Press Enter.

The Table Check Constraints panel is displayed. If the model table contains check constraints, they are listed on this panel.

A In the **Cmd** column, type I (insert), R (repeat) or D (delete). Press HELP for a description of the fields on the panel.

B Press END to display the Table Constraints panel.

6 Press END to display the Create/Alter Table panel.

**To edit the materialized query table options**

1 In the **Edit MQT Options** field, type Y to edit materialized query table options, and press Enter.

The Materialized Query Options panel is displayed (Figure 112).
2 Accept or modify the default attribute values on the panel.

3 In the Help with MQT Text field, type Y to specify additional options.

The Select Generate Text panel is displayed (Figure 113).
Example of creating a table

**Figure 113  Select Generate Text panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Select Text Generate</th>
<th>1 to 5 of 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base table . . . . . . .</td>
<td>QZU.QZUT00_DSC30S2B</td>
<td></td>
</tr>
<tr>
<td>Edit column list . . . . .</td>
<td>N</td>
<td>Y/N</td>
</tr>
<tr>
<td>Default select text from .</td>
<td>NONE</td>
<td>None, Columns, Table</td>
</tr>
<tr>
<td>Full select text . . . . .</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4 Press END twice to return to the Create/Alter Table panel.

**To edit the partitions**

1 In the **Edit partitions** field, type **Y** to edit table partition information.

2 Press **Enter**.

The Table Partitions panel is displayed. If the model table contains partitions, they are listed on the panel.

3 Press END to return to the Create/Alter Table panel.

**To define the organization**

1 In the **Edit organization** field, type **Y** to edit table organization information.

2 Press **Enter**.

3 In the **Select table columns** field, type **Y**.
Example of creating a table

4 On the Select table columns panel, select the columns that you want to include:

A In the **Cmd** column, type 1 beside the column that you want to be first, 2 beside the column that you want to be second, and so on.

B When you have entered an order number for each column that you want to include, press END.

5 Accept or modify the default attribute values on the panel.

6 Press END to return to the Create/Alter Table panel.

**To define the access control**

1 In the **Edit access control** field, type Y to edit table access control information, and press **Enter**.

2 Accept or modify the default attribute values on the panel for row and column access.

3 Press END to return to the Create/Alter Table panel.

**To generate SQL**

1 In the **Generate SQL** field, type Y to generate the SQL that creates the table, and press **Enter**.

The Confirm SQL panel is displayed (Figure 114). This panel shows the statements generated by CATALOG MANAGER based on your specifications.
On the Confirm SQL panel, you can edit and save the SQL to create the object and then execute it.

A *(optional)* From the Command line, issue the SET sqlid command to change the value of the Current SQLID field.

**NOTE**
The ID shown in the Current SQLID field must have the proper authority to perform the specified SQL CREATE statement. If you hold a primary- or secondary-authorization ID that has the proper authority, you can change the Current SQLID to that authorization ID and complete the CREATE. To change the Current SQLID, use the SET command.

B *(optional)* In the Edit Options field, type Y to modify the default values for the options on the Confirm SQL panel. Then, press Enter.

The SQL and Confirm Options panel is displayed. Press END to return to the Confirm SQL panel.

C *(optional)* In the Edit SQL field, type Y or N to invoke an ISPF edit session to edit the SQL statement. Then, press Enter.
Example of creating a table

---

**NOTE**

For views and materialized query tables, CATALOG MANAGER generates a commented-out SET CURRENT SQLID statement in the SQL. (An exception is if the value of the SET CURRENT SQLID to option in the Object Use Options panel is NONE.) For synonyms, CATALOG MANAGER always generates a SET CURRENT SQLID statement in the SQL with the value of the creator for the sqlid.

Press END to save the SQL and return to the Confirm SQL panel.

**D (optional)** In the Save in SQL table field, type A, Y, R, or N to specify whether to save the SQL in the CATALOG MANAGER SQL_Table.

<table>
<thead>
<tr>
<th>To perform this action</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>append the SQL to the SQL in the SQL_Table</td>
<td>A</td>
</tr>
<tr>
<td>save the SQL in the SQL_Table</td>
<td>Y</td>
</tr>
<tr>
<td>replace the SQL in the SQL_Table</td>
<td>R</td>
</tr>
<tr>
<td>discard the SQL</td>
<td>N</td>
</tr>
</tbody>
</table>

**E (optional)** In the Name of saved SQL field, type a name for the SQL.

**F (optional)** In the Save in PDS field, type Y to save the SQL in a member of a partitioned data set (PDS).

The saved SQL uses the ID displayed in the Current SQLID field as the object qualifier. If the SQL is not saved, the ID in the Current SQLID is used only to identify DB2 authority.

**G (optional)** In the PDS(member) field, type the name of the PDS and member.

**H (optional)** Choose one of the following options:

- **(for objects other than native and external SQL procedures)** In the Execute SQL field, type Y to execute the SQL that is displayed on the Confirm SQL panel. Then, press Enter.

  The SQL Progress Indicator panel is displayed. After building the SQL statements that are required to create the table space, CATALOG MANAGER displays the SQL statements in this scrolling panel.

- **(for native and external SQL procedures)** In the Generate worklist field, type Y to generate a batch job to execute the SQL that is displayed on the Confirm SQL panel in a worklist. Then, press Enter.

  The Generate CATALOG MANAGER Worklist Job panel, from which you can build and submit the batch job, is displayed.
Generating DDL to create objects

CATALOG MANAGER provides several commands that generate data definition language (DDL) statements. Unlike the CREATE command (see “Using an existing object as a model to create objects” on page 253), which generates SQL to create objects that are like source objects, the DDL commands generate SQL to create the source objects themselves.

The process for generating DDL to create objects is as follows:

1. Generate an object list.
2. Specify the appropriate DDL command.
3. Generate SQL.

Table 43 describes the different DDL commands.

Table 43  Data definition language commands (Part 1 of 3)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Considerations</th>
<th>Source objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>(from an object list)</td>
<td></td>
<td></td>
<td>AL CK CX DB DT FK FN IM IX MK MQT NP PK PM PR RO SE SG SY TB TC TR TS VW XT</td>
</tr>
<tr>
<td>DDL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(from the Command line)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDL objectCode ownerName.objectName</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>generates DDL to create the objects for which the command was entered</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>for multiple objects, generates individual DDL streams</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The settings for the fields on the following options panels affect the DDL:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>— General Options: Decimal point and SQL string delimiter</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>— SQL and Confirm Options: all fields</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>— SQL Select: all fields</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>— Switches: Define No</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The BATCH keyword is valid for the DDL command. If you use the keyword in</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the command, you must issue the BATCH command to generate JCL. For more</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>information, see “Generating JCL for a job in batch” on page 206.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Generating DDL to create objects

Table 43 Data definition language commands (Part 2 of 3)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Considerations</th>
<th>Source objects</th>
</tr>
</thead>
</table>
| **HDDL** (from an object list) **HDDL** (from the Command line) **HDDL objectCode** **ownerName.objectName** | generates DDL to create the objects for which the command was entered and for the dependent objects | The HDDL command generates the DDL for implicitly created databases, table spaces, tables, and indexes as comments. For most objects, you should not uncomment the DDL. For additional indexes that you have created, the product might have commented out the DDL for the index with the following header in the output:  
```sql
--COMMENTED IMPLICIT
```
If the indexes were created explicitly, uncomment the DDL. | DB MQT NP PR TB TS VW |
| for multiple objects, generates individual DDL streams for each object and its dependents | To generate a single DDL stream for multiple objects and their dependents, issue the **HDDL CONCAT** command against the first object and mark additional objects with the equal (=) sign. | |
| enables you to include GRANT authorizations in the SQL | The BATCH keyword is valid for the HDDL command. If you use the keyword in the command, you must issue the BATCH command to generate JCL. For more information, see “Generating JCL for a job in batch” on page 206. | |
### Table 43  Data definition language commands (Part 3 of 3)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Considerations</th>
<th>Source objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDDL (continued)</td>
<td></td>
<td>The settings for the fields on the following options panels affect the HDDL:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Object Use Options: Include in HDDL and HDESCRIBE and Include in HDDL commit counts</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>— SQL and Confirm Options: all fields</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>— SQL Select: all fields</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Switches: Define No, Cmp &gt; 32k, HDDL Auths, and Build SQLID before GRANT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you specify N for the HDDL Auths switch and later decide to generate GRANT statements without CREATEs, use the HGRANT command. In this way, you can replicate objects on another DB2 subsystem and then grant identical or edited authorizations on the replicated objects. For more information about granting authorizations, see Chapter 10, “Managing authorizations.”</td>
<td></td>
</tr>
<tr>
<td>(from an object list)</td>
<td></td>
<td>generates DDL to create the objects for which the command was entered</td>
<td></td>
</tr>
<tr>
<td>MDDL</td>
<td>applies only a single object type</td>
<td>MDDL is a wait-for-enter command. For more information, see “Issuing Wait-for-Enter commands against multiple objects” on page 59.</td>
<td></td>
</tr>
<tr>
<td>(from the Command line)</td>
<td>for multiple objects of the same object type, generates one DDL stream for all of the objects</td>
<td>The settings for the fields on the following options panels affect the MDDL:</td>
<td></td>
</tr>
<tr>
<td>MDDL objectCode ownerName.objectName</td>
<td></td>
<td>— SQL and Confirm Options: all fields</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>— SQL Select: all fields</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>— The BATCH keyword is not valid for the MDDL command.</td>
<td></td>
</tr>
</tbody>
</table>
Where to go from here

One of the most valuable features of CATALOG MANAGER is the ability to recover object structures and data that have been dropped. Even before performing the drop, you can preview the dependent objects and re-grant the authorizations that will be dropped, assuring yourself that the drop will not have unexpected results.

Chapter 8, “Dropping and recovering objects,” explains how the drop and recovery options in CATALOG MANAGER help you work productively and with a minimum risk of error.
Dropping and recovering objects

This chapter contains the following topics:

Overview .......................................................... 271
Considerations for dropping objects ...................................... 271
  Considerations for dropping a table space ................................. 272
  Considerations for dropping pending changes ......................... 272
  Considerations for dropping a table .............................. 272
  Considerations for dropping an index ............................. 273
Simulating a drop ................................................. 274
Dropping an object .................................................. 274
Recovering an object and its data ........................................ 279
Where to go from here ................................................ 284

Overview

A major concern when executing a drop is whether you can recover the objects if recovery becomes necessary. CATALOG MANAGER addresses this concern by providing drop and drop recovery options that streamline and safeguard the recovery process for dropped object structures and data.

Considerations for dropping objects

Dropping an object can have far-reaching effects that you should carefully consider before you begin the procedure. When you drop an object, you also drop its dependent objects, and you revoke any authorizations on the dropped objects.
Considerations for dropping a table space

When you issue the DROP command on a table space, DB2 drops the table space, any dependent objects, and any pending changes for the table space. CATALOG MANAGER records only the table space and the dependent objects in the Drop Recovery Log.

CATALOG MANAGER can recover table spaces that contain fewer than 26 tables. If you drop a table space that contains more than 26 tables, CATALOG MANAGER cannot recover the table space data.

Considerations for dropping pending changes

You can drop pending changes to table spaces by using one of the following methods:

- Generate a list of table spaces and issue the ALTER command. In the Drop pending DDL field, type Y. DB2 drops the pending changes, but does not drop the table space.

- Generate a list of pending DDL and issue the DROP command. CATALOG MANAGER generates the ALTER TABLESPACE DROP PENDING CHANGES statement. DB2 drops the pending changes, but does not drop the table space.

CATALOG MANAGER does not record the pending changes for the table space in the Drop Recovery Log.

Considerations for dropping a table

Maintaining a database frequently requires changing a table definition, dropping tables, and recovering tables. Some changes to a table definition cannot be achieved with the ALTER TABLE statement. For example, to change some data types, you must drop the table and then redefine it.

In addition, you might decide that you no longer need a table and want to drop it from the database. Before dropping a table, you should consider the implications of removing the table definition from the database. This process involves determining what objects are dependent on the table that you want to drop.
Dropping a table has many implications:

- Referential constraints on the table are dropped.
- Authorizations held on the table are revoked.
- Application plans that use the table are invalidated.
- Table data cannot be recovered unless the table is dropped at the table space level.
- All dependent objects of the table are dropped (views, indexes, and synonyms).
- The table storage space is also affected, as follows:

  — The table space type and manner of creation determines whether the storage space for the table is reclaimed. For example, if the table space containing the table was implicitly created, then the table space is also dropped.

  — When data sets are in a storage group and the table is dropped but the table space is not dropped, the space is reclaimed when the table space is reorganized.

## Considerations for dropping an index

Sometimes an index no longer meets your needs or must be updated to satisfy changing needs. Modifying an index can improve the performance of your applications.

Unless you are simply altering storage allocations for the index, you must first drop the index to be able to re-create and change it. Suppose that you have dropped a simple table space and decide to create a new partitioned table space that uses the same name. One of the new indexes on the new table space must be a partitioning index or must use table-controlled partitioning.

**NOTE**

Unlike dropping table spaces and tables, dropping an index does not cause DB2 to drop any other objects.

Normally, indexes are recovered in the course of recovering a table space or table. However, if you decide to drop an index explicitly, you should enable the Drop Recovery option on the Confirm DROP panel by setting it to Y. Enabling Drop Recovery provides a safeguard in case you need to recover the index definition quickly and easily.
Simulating a drop

By simulating a drop, you can create DROP SQL statements and write them to the Drop Recovery Log without actually dropping the object. The Drop Recovery Log is a DB2 table that is managed by CATALOG MANAGER. It contains all of the information about dropped objects, their dependents, and the SQL to rebuild them that CATALOG MANAGER needs to invoke the DSN1COPY utility to re-create the structures and authorizations.

1. Set the Drop switch to N. For information, see “Setting CATALOG MANAGER switches” on page 95.

   When the Drop switch is set to N, CATALOG MANAGER performs all of the actions for a drop except the actual execution. Then, CATALOG MANAGER issues a message stating that the drop was not executed.

2. Follow the steps in “Dropping an object” on page 274.

Dropping an object

The following procedure describes how to drop a table space.

1. Take a full-image copy of the table space that you intend to drop.

   **NOTE**
   
   CATALOG MANAGER does not permit you to perform a drop if a full-image copy of the object is not present in the SYSCOPY catalog table.

2. Generate a table space list that contains the table space that you want to drop. For information, see “Generating lists in CATALOG MANAGER” on page 47.

3. Type DROP in the Cmd column beside the table space to be dropped.

   DROP is a wait-for-enter command that can be issued for multiple objects on the list. (For information about dropping multiple objects, see “Issuing Wait-for-Enter commands against multiple objects” on page 59.)

4. Press Enter.

   The Confirm DROP panel is displayed (Figure 115), providing several actions and options that relate only to the drop and drop recovery functions.
Figure 115  Confirm DROP panel for DROP TABLE SPACE procedure

DEFF-R  Confirm DROP                         1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE

Current SQLID. . . . . . . .  RDACRJ
Edit Options . . . . . . . .  N   Y/N Modify SQL processing options
Generate Dependency List . .  N   Y/N Only for a single drop
Edit SQL . . . . . . . . . .  N   Y/N Edit SQL before executing
Save in SQL table. . . . . .  N   A/Y/R/N A/Y-Append, R-Replace
Name of saved SQL . . . . .  20110121_115552
Save in PDS. . . . . . . . .  N   Y/N Save SQL in PDS
PDS(member) . . . . . . . .  
Remove DROP RESTRICT . . .  N   Y/N To remove RESTRICT ON DROP
Drop Recovery. . . . . . . .  N   Y/N For DDL Only, NOT DATA
Log Image Copies . . . . . .  N   Y/N Ignored unless Drop Recover is Y
Execute SQL. . . . . . . . .  N   Y/N Execute the SQL
Generate worklist. . . . . .  N   Y/N Execute the SQL in Batch
----------------------  SQL  ----------------------
DROP TABLESPACE QZUDAC.QZUS01AC ;
****************************************************************************** Bottom of data ******************************************************************************

5  On the Confirm DROP panel, you can edit and save the SQL to drop the object and then execute it.

A  (optional)  From the Command line, issue the SET sqlid command to change the value of the Current SQLID field.

NOTE
The ID shown in the Current SQLID field must have the proper authority to perform the specified SQL DROP statement. If you hold a primary- or secondary-authorization ID that has the proper authority, you can change the Current SQLID to that authorization ID and complete the drop. To change the Current SQLID, use the SET command.

B  (optional)  In the Edit options field, type Y to modify the default values for the options on the Confirm DROP panel. Then, press Enter.

The SQL and Confirm Options panel is displayed. Press END to return to the Confirm DROP panel.

C  (optional)  In the Generate Dependency List field, type Y or N to view the list of dependent objects that will be dropped with the table space.
Dropping an object

NOTE
The Generate Dependency List option is not available when you drop multiple objects.

CATALOG MANAGER displays the Drop Dependency List and includes a list of the dependent objects that will be dropped (Figure 116). Review the list to verify that you want to drop all of the dependent objects.

Figure 116 Drop Dependency List

D (optional) In the Edit SQL field, type Y or N to invoke an ISPF edit session to edit the SQL statement. Then, press Enter.

NOTE
For native SQL procedures, the following statement is generated:

```
ALTER PROCEDURE procedureName
  DROP VERSION versionName;
```

This statement enables you to drop a single version of a procedure. To drop all versions of the procedure, edit the statement as follows:

```
DROP PROCEDURE procedureName RESTRICT;
```

Press END to save the SQL and return to the Confirm DROP panel.
E (optional) In the **Save in SQL table** field, type A, Y, R, or N to specify whether to save the SQL in the CATALOG MANAGER SQL_Table.

<table>
<thead>
<tr>
<th>To perform this action</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>append the SQL to the SQL in the SQL_Table</td>
<td>A</td>
</tr>
<tr>
<td>save the SQL in the SQL_Table</td>
<td>Y</td>
</tr>
<tr>
<td>replace the SQL in the SQL_Table</td>
<td>R</td>
</tr>
<tr>
<td>discard the SQL</td>
<td>N</td>
</tr>
</tbody>
</table>

F (optional) In the **Name of saved SQL** field, type a name for the SQL.

G (optional) In the **Save in PDS** field, type Y to save the SQL in a member of a partitioned data set (PDS).

The saved SQL uses the ID displayed in the **Current SQLID** field as the object qualifier. If the SQL is not saved, the ID in the **Current SQLID** is used only to identify DB2 authority.

H (optional) In the **PDS(member)** field, type the name of the PDS and member.

I (optional) In the **Remove DROP RESTRICT** field, type Y to have CATALOG MANAGER analyze each DROP command of a database, table space, or table.

If a table includes the DROP RESTRICT attribute, the product generates an ALTER TABLE DROP RESTRICT ON DROP command before the DROP command. However, the Drop Recovery Log will not include the DROP RESTRICT attribute in the CREATE TABLE statement.

J (optional) In the **Drop Recovery** field, type Y to log the dropped objects to the Drop Recovery Log. This option generates the DDL to re-create the object structures and DB2 authorizations.

K (optional) In the **Log Image Copies** field, type Y to save image copy information so that you can recover a copy of the data for the table space and its dependent objects.

**WARNING**

You must enable both the **Drop Recovery** and **Log Image Copies** options to perform the steps to recover data. If you set the **Drop Recovery** value to Y and the **Log Image Copies** value to N, you can recover the object structures but not the data.

L (optional) In the **Generate worklist** field, type Y to generate a batch job to execute the SQL in a worklist.
The Generate CATALOG MANAGER Worklist Job panel, from which you can build and submit the batch job, is displayed.

**NOTE**

DB2 allows some objects to be dropped in batch mode. To prevent possible problems with catalog contention while the worklist is executed, CATALOG MANAGER generates COMMIT statements between DROP statements.

*(optional)* In the Execute SQL field, type Y to execute the SQL displayed on the Confirm DROP panel. Then, press Enter.

The SQL Progress Indicator panel is displayed. After building the SQL statements that are required to recover the table space, CATALOG MANAGER displays the SQL statements in this scrolling panel.

After executing the drop, CATALOG MANAGER displays the Table Space List panel with the marker *DROP beside the dropped table space name (Figure 117). The marker is removed when the list is refreshed.

- If you have set the value of the Drop switch to N (see “Simulating a drop” on page 274), CATALOG MANAGER displays DROP NOT DONE in the short message field to alert you that the drop was not actually executed.

- If the drop was executed, CATALOG MANAGER displays the return code from the DROP command in the short message field.

---

**Figure 117** Table Space List panel after drop

<table>
<thead>
<tr>
<th>Tablespace</th>
<th>Owner</th>
<th>Segsz</th>
<th>Bpool</th>
<th>Prts</th>
<th>Tbls</th>
<th>ActivPg</th>
<th>Status</th>
<th>Enc</th>
<th>Ty</th>
</tr>
</thead>
<tbody>
<tr>
<td>*drop AC.QZUS01AC</td>
<td>ASUQA</td>
<td>0</td>
<td>BP0</td>
<td>4</td>
<td>1</td>
<td>23K</td>
<td>A</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS02AC</td>
<td>ASUQA</td>
<td>0</td>
<td>BP0</td>
<td>4</td>
<td>1</td>
<td>23K</td>
<td>A</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS03AC</td>
<td>ASUQA</td>
<td>4</td>
<td>BP0</td>
<td>1</td>
<td>1</td>
<td>5040</td>
<td>A</td>
<td>E</td>
<td>G</td>
</tr>
<tr>
<td>QZUDAC.QZUS04AC</td>
<td>ASUQA</td>
<td>4</td>
<td>BP0</td>
<td>0</td>
<td>1</td>
<td>24K</td>
<td>A</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS05AC</td>
<td>ASUQA</td>
<td>8</td>
<td>BP0</td>
<td>0</td>
<td>1</td>
<td>24K</td>
<td>A</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS06AC</td>
<td>ASUQA</td>
<td>16</td>
<td>BP0</td>
<td>0</td>
<td>1</td>
<td>24K</td>
<td>A</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS07AC</td>
<td>ASUQA</td>
<td>32</td>
<td>BP0</td>
<td>0</td>
<td>1</td>
<td>24K</td>
<td>A</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS08AC</td>
<td>ASUQA</td>
<td>64</td>
<td>BP0</td>
<td>0</td>
<td>1</td>
<td>24K</td>
<td>A</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS09AC</td>
<td>ASUQA</td>
<td>16</td>
<td>BP0</td>
<td>2</td>
<td>1</td>
<td>307K</td>
<td>A</td>
<td>E</td>
<td>G</td>
</tr>
<tr>
<td>QZUDAC.QZUS10AC</td>
<td>ASUQA</td>
<td>4</td>
<td>BP0</td>
<td>0</td>
<td>1</td>
<td>130</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS11AC</td>
<td>ASUQA</td>
<td>16</td>
<td>BP0</td>
<td>0</td>
<td>2</td>
<td>146</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS12AC</td>
<td>ASUQA</td>
<td>0</td>
<td>BP0</td>
<td>4</td>
<td>1</td>
<td>1440</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS13AC</td>
<td>ASUQA</td>
<td>64</td>
<td>BP0</td>
<td>0</td>
<td>2</td>
<td>540</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS14AC</td>
<td>ASUQA</td>
<td>0</td>
<td>BP0</td>
<td>4</td>
<td>1</td>
<td>720</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>QZUDB1.QZUS01B1</td>
<td>ASUQA</td>
<td>4</td>
<td>BP0</td>
<td>0</td>
<td>1</td>
<td>8221</td>
<td>A</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>QZUDB1.QZUS02B1</td>
<td>ASUQA</td>
<td>4</td>
<td>BP32K</td>
<td>0</td>
<td>1</td>
<td>1005</td>
<td>A</td>
<td>E</td>
<td></td>
</tr>
</tbody>
</table>
Recovering an object and its data

The following procedure describes how to recover a table space structure that was dropped by one of the following BMC products and how to recover the data in the table space:

- CATALOG MANAGER
- ALTER
- CHANGE MANAGER

**NOTE**
You must have enabled the **Drop Recovery** and **Log Image Copies** options on the Confirm DROP panel to perform this procedure. For more information, see “Dropping an object” on page 274.

**To recover the structure and authorizations**

1. From the Command line of the Primary Menu panel or a list panel, type **DROPR TS** to display a list of dropped table spaces.

   **NOTE**
   To display a list of all dropped objects, enter **DROPR ALL**.

2. Press Enter.

   The Drop Recovery List panel is displayed (Figure 118). This list comprises a subset of the Drop Recovery Log.
Recovering an object and its data

Figure 118  Drop Recovery List panel for dropped table spaces

<table>
<thead>
<tr>
<th>DEFF-R</th>
<th>Drop Recovery List</th>
<th>Command ===&gt;</th>
<th>Scroll ===&gt; CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CMD will show commands for this list. Type command and press ENTER. Enter S to select the object to be recovered.</td>
<td></td>
</tr>
<tr>
<td>TABLESPACE</td>
<td></td>
<td>Cmd Date Time AuthId Product Type Name</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>---v----1----v----2----v----3----v----4----v----5----v----6----v----7----v---</td>
<td></td>
</tr>
<tr>
<td>2011-02-01 14.43 RDAPXB2 ACT010 TS DBXNAUT.SBXNCOLO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-19 16.25 RDAPXB2 ACT010 TS PLBDBA1.PLBLA01TO2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-19 14.09 RDAPXB2 ACT010 TS PLBDBA1.PLBLATR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-17 14.16 RDAPXB2 ACT010 TS PXBQXBP_HASHPXB1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-10 16.33 SKH2 AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-10 16.13 SKH2 AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-10 16.01 SKH2 AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-10 15.39 SKH2 AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-10 15.27 SKH2 AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-10 15.25 SKH2 AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-08 12.58 RDAMSL AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-08 11.25 RDAMSL AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-07 18.21 RDAMSL AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-07 17.44 RDAMSL AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-07 16.58 SKH AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011-01-07 16.44 SKH AEX010 TS MXCDBG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3  (optional) Produce the SQL that is required to create the table space and its dependents.

A Type 2SQL in the Cmd column beside the table space to be recovered.

B Press Enter.

The Confirm SQL panel is displayed, from which you can edit the SQL or save it in the SQL Table or a PDS. The 2SQL command works similarly to the HDDL command (see “Generating DDL to create objects” on page 267).

C Press END.

D Go to step 1.

4 Type S in the Cmd column beside the table space to be recovered, and then press Enter.

The Recovery Statements panel is displayed (Figure 119), which shows which objects will be recovered with the table space.
Recovering an object and its data

Chapter 8  Dropping and recovering objects  281

Figure 119  Recovery Statements panel

CATALOG MANAGER automatically excludes the statements that are shown in Table 44. These statements are displayed on the Recovery Statements panel, but are marked with X for exclusion from processing.

Table 44  Statements excluded from object recovery

<table>
<thead>
<tr>
<th>Exclusion</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>REC DATA</td>
<td>You must recover objects before recovering data.</td>
</tr>
<tr>
<td>REC LRBA</td>
<td>This is a comment which points to the log rba.</td>
</tr>
<tr>
<td>SYNONYM</td>
<td>Exclusion enables you to change the SQLID for synonyms.</td>
</tr>
</tbody>
</table>

5 To mark additional objects for exclusion from recovery, type X in the **Cmd** column beside each object.

**NOTE**

You can mark all indexes for exclusion by typing X on the **Command** line.

6 Type **D** or **S** in the **Cmd** column beside statements for which you want to display a detailed log entry. **Figure 120** shows a Describe Audit Log Entry panel. Press END to display the Recovery Statements Panel.
7 In the **Execute recovery statements now** field, type `Y` to recover the object structures and authorizations. Press Enter.

The Recovery Statements panel is refreshed to show the results of the recovery operation. An **OK** message in the **Cmd** column indicates successful completion of the statement. The object structures and authorizations have now been recovered.

### To recover the data

1 In the Recovery Statements panel, type **STOP** in the **Cmd** column of the **REC DATA** text line.

    **NOTE**

You must **STOP** the table space because the DSN1COPY utility requires exclusive use of the data set.

2 Press Enter.

The DB2 Commands panel is displayed.

3 Press Enter to execute the command.

CATALOG MANAGER sends the command to DB2.
4 Press END.

The Drop Recovery List is displayed.

5 Type S in the **Cmd** column beside the table space to be recovered, and then press **Enter**.

The Recovery Statements panel is displayed.

6 Type **RECOVER** in the **Cmd** column of the **REC DATA** text line. Press **Enter**.

---

**NOTE**

If several incremental image copies exist, or if the table space had several partitions, several **REC DATA** text lines might be displayed in the Recovery Statements panel. To recover data, you need to type **RECOVER** for only one **REC DATA** line.

CATALOG MANAGER recovers only to the last image copy. If you also want to recover changes from the log, you must know the **to logrba**. **REC LRBA** provides the **to logrba**. Recovery from the log is only possible if the OBIDs did not change between dropping and recreating the objects.

CATALOG MANAGER invokes the DSN1COPY utility to recover the data that was stored in the dependent tables. The DSN1COPY Droprecovery Utility panel is displayed. The Utility Options section of the panel shows each option that you can set when using DSN1COPY. Press HELP for a description of these options.

7 In the **JCL Dataset** field, verify that the specified JCL data set name is allocated and that this is the correct data set.

8 In the **Set JCL Options** field, type Y to set your options for generating JCL.

The JCL Generation Options panel is displayed. Press END to return to the DSN1COPY Droprecovery Utility panel.

9 In the **Build Job** field, type Y to build the JCL. Press **Enter**.

10 In the **Edit Dataset** field, type Y to edit the data set. Press **Enter**.

The JCL data set is displayed in the ISPF edit panel. After editing, press END to display the DSN1COPY Droprecovery Utility panel.

11 In the **Submit** field, type Y to submit the JCL. Press **Enter**.

The batch job must be completed before you can proceed.

12 When the batch job is completed, press END to display the Recovery Statements panel.
13 To restart the table space, type START in the Cmd column of the REC DATA text line. Press Enter.

The DB2 Commands panel is displayed.

14 Press Enter to execute the command.

CATALOG MANAGER sends the command to DB2.

15 Press END.

The Drop Recovery List is displayed.

16 Type S in the Cmd column beside the table space to be recovered, and then press Enter.

The Recovery Statements panel is displayed.

17 To recover indexes, complete the following steps:

---

**NOTE**

To recover more than one index in the same table space, create a table space (TS) list, enter RECOVER IX ALL in the Cmd column beside the table space name.

---

A Remove the X notations from any indexes that you marked for exclusion.

B In the Execute recovery statements now field, type Y.

C Press Enter.

The table space and data recovery are complete.

---

**Where to go from here**

With CATALOG MANAGER you can easily generate and submit JCL for BMC and IBM utility jobs. Chapter 9, “Generating utility jobs,” shows you how.
Generating utility jobs

This chapter contains the following topics:

Overview .................................................. 285
Available utilities ...................................... 286
  BMC products ........................................... 286
  IBM DB2 utilities ................................... 287
  Issuing utility commands ............................ 288
Utility profiles .......................................... 288
  Creating a utility profile for a single utility .... 289
  Creating a utility profile for multiple utilities ... 294
  Creating a utility profile from an existing profile . 296
  Editing a utility profile .............................. 298
  Working with the last-used utility profile ....... 299
  Switching utility profiles ............................ 300
TEMPLATE and LISTDEF control statements ........ 300
  Available utilities for TEMPLATE and LISTDEF statements ........................................ 301
  Incorporating TEMPLATE and LISTDEF statements .................................................. 301
Where to go from here .................................. 304

Overview

This chapter describes how to generate BMC and IBM DB2 utility jobs by using CATALOG MANAGER.

Utility options enable you to customize Job Control Language (JCL) parameters to meet your installation requirements and office standards. For convenience, after setting default values for utility options, you can save the values in a utility profile for repeated use.
Available utilities

In CATALOG MANAGER, you can initiate processing for BMC products and IBM utilities.

BMC products

In CATALOG MANAGER, you can initiate processing for BMC products by using the commands listed in Table 45. For information about issuing commands in CATALOG MANAGER, see “Listing and executing commands” on page 56.

Table 45 Commands to invoke BMC products

<table>
<thead>
<tr>
<th>CATALOG MANAGER command</th>
<th>BMC product</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCCHECK</td>
<td>CHECK PLUS for DB2</td>
</tr>
<tr>
<td>BMCCHECK INDEX</td>
<td>CHECK PLUS for DB2</td>
</tr>
<tr>
<td>BMCCHECK IX</td>
<td>CHECK PLUS for DB2</td>
</tr>
<tr>
<td>BMCCHTS</td>
<td>CHECK PLUS for DB2</td>
</tr>
<tr>
<td>BMCCOPY</td>
<td>COPY PLUS for DB2</td>
</tr>
<tr>
<td>BMCCOPY INDEX</td>
<td>COPY PLUS for DB2</td>
</tr>
<tr>
<td>BMCCOPY IX</td>
<td>COPY PLUS for DB2</td>
</tr>
<tr>
<td>BMCCOPY LOADPLUS for DB2</td>
<td></td>
</tr>
<tr>
<td>BMCREBUILD</td>
<td>RECOVER PLUS for DB2</td>
</tr>
<tr>
<td>BMCREBUILD INDEX</td>
<td>RECOVER PLUS for DB2</td>
</tr>
<tr>
<td>BMCREBUILD IX</td>
<td>RECOVER PLUS for DB2</td>
</tr>
<tr>
<td>BMCRECOVER</td>
<td>RECOVER PLUS for DB2</td>
</tr>
<tr>
<td>BMCRECOVER INDEX</td>
<td>RECOVER PLUS for DB2</td>
</tr>
<tr>
<td>BMCRECOVER IX</td>
<td>RECOVER PLUS for DB2</td>
</tr>
<tr>
<td>BMCREORG</td>
<td>REORG PLUS for DB2</td>
</tr>
<tr>
<td>BMCREORG INDEX</td>
<td>REORG PLUS for DB2</td>
</tr>
<tr>
<td>BMCREORG IX</td>
<td>REORG PLUS for DB2</td>
</tr>
<tr>
<td>BMCSTATS</td>
<td>BMCSTATS (component of the BMC DASD MANAGER PLUS for DB2 product)</td>
</tr>
<tr>
<td>BMCSTATS INDEX</td>
<td>BMCSTATS (component of the BMC DASD MANAGER PLUS for DB2 product)</td>
</tr>
<tr>
<td>BMCSTATS IX</td>
<td>BMCSTATS (component of the BMC DASD MANAGER PLUS for DB2 product)</td>
</tr>
<tr>
<td>BMCUNLOAD</td>
<td>UNLOAD PLUS for DB2</td>
</tr>
</tbody>
</table>

CATALOG MANAGER Help panels and Appendix E, “Commands,” provide more information about using these commands. For more information about a particular BMC product, see the reference manual for that product.
IBM DB2 utilities

To initiate processing for IBM DB2 utilities, use the commands listed in Table 46. For information about issuing commands in CATALOG MANAGER, see “Listing and executing commands” on page 56.

Table 46  Commands to invoke IBM DB2 utilities (Part 1 of 2)

<table>
<thead>
<tr>
<th>CATALOG MANAGER command</th>
<th>IBM DB2 utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK</td>
<td>CHECK DATA</td>
</tr>
<tr>
<td>CHECK INDEX</td>
<td>CHECK INDEX</td>
</tr>
<tr>
<td>CHECK IX</td>
<td>CHECK INDEX</td>
</tr>
<tr>
<td>CHKI</td>
<td>CHECK INDEX</td>
</tr>
<tr>
<td>COPY</td>
<td>COPY</td>
</tr>
<tr>
<td>COPY INDEX</td>
<td>COPY</td>
</tr>
<tr>
<td>COPY IX</td>
<td>COPY</td>
</tr>
<tr>
<td>COPYTOCOPY</td>
<td>COPYTOCOPY</td>
</tr>
<tr>
<td>COPYTOCOPY INDEX</td>
<td>COPYTOCOPY</td>
</tr>
<tr>
<td>COPYTOCOPY IX</td>
<td>COPYTOCOPY</td>
</tr>
<tr>
<td>DSN1COMP</td>
<td>DSN1COMP</td>
</tr>
<tr>
<td>DSN1COPY</td>
<td>DSN1COPY</td>
</tr>
<tr>
<td>EXEC</td>
<td>EXEC SQL</td>
</tr>
<tr>
<td>LOAD</td>
<td>LOAD</td>
</tr>
<tr>
<td>MERGECOPY</td>
<td>MERGECOPY</td>
</tr>
<tr>
<td>MODIFY</td>
<td>MODIFY RECOVERY</td>
</tr>
<tr>
<td>MODIFYRECOVERY</td>
<td>MODIFY RECOVERY</td>
</tr>
<tr>
<td>MODIFYSTATISTIC</td>
<td>MODIFY STATISTICS</td>
</tr>
<tr>
<td>QUIESCE</td>
<td>QUIESCE</td>
</tr>
<tr>
<td>REBUILD</td>
<td>REBUILD INDEX</td>
</tr>
<tr>
<td>REBUILD INDEX</td>
<td>REBUILD INDEX</td>
</tr>
<tr>
<td>REBUILD IX</td>
<td>REBUILD INDEX</td>
</tr>
<tr>
<td>RECOVER</td>
<td>RECOVER</td>
</tr>
<tr>
<td>RECOVER INDEX</td>
<td>RECOVER</td>
</tr>
<tr>
<td>RECOVER IX</td>
<td>RECOVER</td>
</tr>
<tr>
<td>REORG</td>
<td>REORG TABLESPACE</td>
</tr>
<tr>
<td>REORG INDEX</td>
<td>REORG INDEX</td>
</tr>
<tr>
<td>REORG IX</td>
<td>REORG INDEX</td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT</td>
</tr>
<tr>
<td>REPORT INDEX</td>
<td>REPORT</td>
</tr>
<tr>
<td>REPORT IX</td>
<td>REPORT</td>
</tr>
<tr>
<td>RUNSTATS</td>
<td>RUNSTATS</td>
</tr>
<tr>
<td>RUNSTATS INDEX</td>
<td>RUNSTATS</td>
</tr>
<tr>
<td>RUNSTATS IX</td>
<td>RUNSTATS</td>
</tr>
</tbody>
</table>
Issuing utility commands

Because the utility commands are Wait-for-Enter (WFE) commands, in one operation you can issue a utility command against more than one listed object. CATALOG MANAGER then builds one JCL job that includes all of the utility commands. The number of items or instances allowed for a utility depends on the number of control statements that the utility can execute. For example, the BMC utilities allow one complete control statement for each job step, but other utilities can combine a set of control statements within a single job step. In either case, CATALOG MANAGER allows a maximum of 999 control statement sets to be executed.

For more information about issuing commands in CATALOG MANAGER, see “Issuing Wait-for-Enter commands against multiple objects” on page 59.

Utility profiles

A utility profile is a file that contains customized specifications for the syntax of a utility. Using utility profiles saves you time and helps avoid user errors because you avoid having to specify the syntax each time that you generate a job. Using utility profiles is not required; however, it is strongly recommended by BMC.

CATALOG MANAGER creates the following types of utility profiles:

- site profiles, which apply to all users
- user profiles, which are available to the users who create them

| Table 46  Commands to invoke IBM DB2 utilities (Part 2 of 2) |
|-----------------|-----------------|
| **CATALOG MANAGER command** | **IBM DB2 utility** |
| STOSPACE        | STOSPACE        |
| UNLOAD          | UNLOAD          |

CATALOG MANAGER Help panels and Appendix E, “Commands,” provide more information about using these commands. For information about the IBM DB2 utilities, see the IBM documentation.
Minimal setup is required to enable you to use utility profiles:

- Allocate a data set in which to save the utility profiles by using the following criteria:
  
  - Record format = VB (variable blocked)
  - Record length = 255

- Specify the data set in CATALOG MANAGER options, as follows:
  
  - Specify the site profile data set in the TDSN installation option of the CATALOG MANAGER installation options. The default value of the TDSN option is blank. The data set can be either sequential or partitioned. If the data set is partitioned, ensure that you include the member name in the data set. For more information, see the installation guide.

  - Specify the user profile data set on the Datasets panel. For more information, see “Setting data set options” on page 88.

BMC recommends that you save all utility profiles in members of the same data set.

Creating a utility profile for a single utility

This procedure describes how to create a utility profile and generate a utility job for a single utility.

To start the utility

1. Generate an object list.

   For information, see “Generating lists in CATALOG MANAGER” on page 47.

2. For a list of the commands that invoke specific utilities, see Table 45 and Table 46. Alternatively, you can display a list of the commands and the valid list types by typing CMD on the Command line.

3. Issue the command to invoke the applicable utility, as follows:

   - To issue the command against all objects on the object list, type command ALL on the Command line.

   - To issue the command against specific objects on the object list, type the command in the Cmd column beside the specified objects.
Press Enter.

The Utility List panel is displayed. Figure 121 shows the Utility List panel for the REORG PLUS utility.

Figure 121 Utility List panel

To set JCL and worklist options

The fields on the top half of the Utility List panel enable you to specify the JCL and worklist options.

1 In the JCL Dataset field, verify the name of the data set where the generated JCL is stored.

You specify this data set name on the Datasets panel. If this is a partitioned data set, you must specify a member name. For more information, see “Setting data set options” on page 88.

To specify a different data set name for only the job that you are building, type the data set name in the JCL Dataset field. The change does not persist for subsequent jobs.

2 In the User Profile Dsn field, verify the data set and member for user utility profiles.
3 In the Set JCL options field, type Y to view and modify the JCL options panels, and then press Enter.

4 After specifying the JCL options, press END to display the Utility List panel.

5 In the Utility ID field, type a unique identification for this utility operation.

You can include from 1 to 16 characters. If you do not enter a value for this field, CATALOG MANAGER substitutes the SSID for the Utility ID in the generated JCL.

The Utility ID field can contain variable names which will be changed to the variable value in the generated JCL. Table 47 shows valid variables for the Utility ID.

Table 47 Utility ID variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;UTIL</td>
<td>the utility name, such as REORG PLUS or LOAD PLUS</td>
</tr>
<tr>
<td>&amp;ACTUDB&lt;sup&gt;a&lt;/sup&gt;</td>
<td>the database of the first table space that is selected</td>
</tr>
<tr>
<td>&amp;ACTUITS&lt;sup&gt;a&lt;/sup&gt;</td>
<td>the first table space that is selected</td>
</tr>
<tr>
<td>&amp;ZUSER</td>
<td>the ISPF variable for User ID</td>
</tr>
<tr>
<td>&amp;ZDATE</td>
<td>the ISPF variable for Date</td>
</tr>
</tbody>
</table>

<sup>a</sup> &ACTUITS and &ACTUDB might not be meaningful if the objects addressed by the utility are indexes or stogroups.

To set the utility options values

The bottom section of the Utility List panel includes the numbered utility statements and the commands that can be used to edit the utility statements and to display utility profiles.

1 In the Cmd column beside each utility statement that you want to customize, type ED and press Enter.

For a list of the commands that you can enter in the Cmd column beside the applicable utility statements, see “Utility list commands” on page 464.

---

**NOTE**

Except for edits that are specific to DB2 objects, edits are propagated to subsequent objects on the list.

---

A utility panel is displayed. You can specify option values by using the fields below the dotted line on the panel (Figure 122).
Creating a utility profile for a single utility

2 (optional) You can enter another Utility Id value on the utility panel. If you do so, the value on the utility panel takes precedence over the Utility Id value that you entered on the Utility List panel. If you do not enter a Utility Id value on either panel, CATALOG MANAGER uses your SSID for the utility ID in the generated JCL.

3 Define values for the options as needed. Additional options panels might be available for most utilities.

---

**NOTE**

Panel help and field-level help are available.

- For panel help, place the cursor on the Command line, and then press HELP.
- For field-level help, place the cursor on any option field, and then press HELP.

For information about specific utility products, see the BMC or IBM documentation.

4 After setting the appropriate values for all selected utility statements, on the Command line, type PROFILE SAVEAS profileID to save the values as a utility profile.

You can use a maximum of 64 characters, including spaces, in the profile ID. The characters <, >, ?, and / are not permitted.
5 Press Enter.

CATALOG MANAGER displays a message to confirm that your settings have been saved as specified.

6 Press END to display the options panels for subsequent statements or to display the Utility List panel.

Edited statements are marked as Edited in the Status column (Figure 123). The profile ID is displayed in the Profile column.

---

NOTE

Profile IDs are truncated on the Utility List panel. To display the entire profile ID, place your cursor in the field and press ZOOM (F4).

---

Figure 123 Utility List panel after editing utility statements

<table>
<thead>
<tr>
<th>Command ====&gt;</th>
<th>Utility List 1 to 6 of 6</th>
<th>Scroll ====&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCL Dataset. . . . . . .</td>
<td>'RDACRJ.BMCCAT.JCL(REORPLUS)'</td>
<td></td>
</tr>
<tr>
<td>User Profile Dsn . . . . .</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Set JCL options. . . . . . . N Y/N - Change options for creating JCL
Build Job. . . . . . . . . . Y Y/N - Create JCL, save in dataset
Edit Dataset . . . . . . . . Y Y/N - Edit JCL dataset
Submit . . . . . . . . . . . . N Y/N - Submit JCL dataset
Utility ID . . . . . . . . . . RBLDPLUS

<table>
<thead>
<tr>
<th>Cmd Status</th>
<th>St#</th>
<th>Utility</th>
<th>Typ</th>
<th>Name</th>
<th>Part</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edited</td>
<td>1</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD34.QZUS0134</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD34.QZUS0234</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD35.QZUS0135</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD35.QZUS0235</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD35.QZUS0335</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD35.QZUS0435</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

******************************* Bottom of data ********************************

To read the status column

After you modify the utility statements and utility options, the Status column for each statement reflects the actions that you have taken as shown in Table 48.
Creating a utility profile for multiple utilities

To build and submit the job

After customizing the utility statements and utility options values, you can build and submit the utility generation job.

1. In the Build Job field, type Y to build the JCL by using the specified parameters.

2. To view and edit the generated JCL, complete the following steps:
   A. In the Edit Dataset field, type Y.
   B. Press Enter.
      After the job is built, the JCL data set is displayed in an ISPF edit panel.
   C. Verify that the JCL is correct or edit if necessary.
   D. Press END to display the Utility List panel.

3. In the Submit field, type Y to execute the JCL.

4. Press Enter.

Creating a utility profile for multiple utilities

This procedure describes how to create a utility profile and generate a utility job for multiple utilities.

1. Generate an object list.

   For information, see “Generating lists in CATALOG MANAGER” on page 47.
2 Issue the UTILITY (UTILITY) command, as follows:

- To issue the command against all objects on the object list, on the Command line, type UTILITY ALL.

- To issue the command against specific objects on the object list, type UTILITY in the Cmd column beside the source objects.

3 Press Enter.

The Utility Selections panel is displayed. Figure 124 shows the Utility Selections Panel for table spaces.

Figure 124 Utility Selections panel

<table>
<thead>
<tr>
<th>BMC Utilities</th>
<th>IBM Utilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCSTATS . . . . . .</td>
<td>CHECK DATA . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (Data). . .</td>
<td>CHECK INDEX. . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (IX). . .</td>
<td>COPY . . . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (TS). . .</td>
<td>MODIFY RECOVERY. . .</td>
</tr>
<tr>
<td>BMCSTATS . . . . . .</td>
<td>MODIFY STATISTICS. . .</td>
</tr>
<tr>
<td>CHECK PLUS (Data). . .</td>
<td>DSNICOMP . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (IX). . .</td>
<td>DSNICOPY . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (TS). . .</td>
<td>EXEC SQL . . . . .</td>
</tr>
<tr>
<td>BMCSTATS . . . . . .</td>
<td>LOAD . . . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (Data). . .</td>
<td>MERGECOPY . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (IX). . .</td>
<td>MODIFY RECOVERY. . .</td>
</tr>
<tr>
<td>CHECK PLUS (TS). . .</td>
<td>MODIFY STATISTICS. . .</td>
</tr>
<tr>
<td>BMCSTATS . . . . . .</td>
<td>DSNICOMP . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (Data). . .</td>
<td>DSNICOPY . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (IX). . .</td>
<td>EXEC SQL . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (TS). . .</td>
<td>LOAD . . . . . . .</td>
</tr>
<tr>
<td>BMCSTATS . . . . . .</td>
<td>MERGECOPY . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (Data). . .</td>
<td>MODIFY RECOVERY. . .</td>
</tr>
<tr>
<td>CHECK PLUS (IX). . .</td>
<td>MODIFY STATISTICS. . .</td>
</tr>
<tr>
<td>CHECK PLUS (TS). . .</td>
<td>DSNICOMP . . . . .</td>
</tr>
<tr>
<td>BMCSTATS . . . . . .</td>
<td>DSNICOPY . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (Data). . .</td>
<td>EXEC SQL . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (IX). . .</td>
<td>LOAD . . . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (TS). . .</td>
<td>MERGECOPY . . . . .</td>
</tr>
<tr>
<td>BMCSTATS . . . . . .</td>
<td>MODIFY RECOVERY. . .</td>
</tr>
<tr>
<td>CHECK PLUS (Data). . .</td>
<td>MODIFY STATISTICS. . .</td>
</tr>
<tr>
<td>CHECK PLUS (IX). . .</td>
<td>DSNICOMP . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (TS). . .</td>
<td>DSNICOPY . . . . .</td>
</tr>
<tr>
<td>BMCSTATS . . . . . .</td>
<td>EXEC SQL . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (Data). . .</td>
<td>LOAD . . . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (IX). . .</td>
<td>MERGECOPY . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (TS). . .</td>
<td>MODIFY RECOVERY. . .</td>
</tr>
<tr>
<td>BMCSTATS . . . . . .</td>
<td>MODIFY STATISTICS. . .</td>
</tr>
<tr>
<td>CHECK PLUS (Data). . .</td>
<td>DSNICOMP . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (IX). . .</td>
<td>DSNICOPY . . . . .</td>
</tr>
<tr>
<td>CHECK PLUS (TS). . .</td>
<td>EXEC SQL . . . . .</td>
</tr>
<tr>
<td>BMCSTATS . . . . . .</td>
<td>LOAD . . . . . . .</td>
</tr>
</tbody>
</table>

4 Select utilities in a desired order by typing a number between 0 and 8 beside each utility. Use ISPF scrolling commands or function keys to move up and down the list.

5 On the Command line, type PROFILE SAVEAS profileID to save the values as a utility profile.

You can use a maximum of 64 characters, including spaces, in the profile ID. The characters <, >, ?, and / are not permitted.
6 Press END to display the Utility List panel.

The utilities are displayed in the Utility field in the numeric order that you specified.

To determine JCL and worklist options, set values for the utility options, and execute the JCL, see “Creating a utility profile for a single utility” on page 289.

Creating a utility profile from an existing profile

To create a profile that is similar to a profile for the same utility type, complete the following steps:

1 Generate an object list.

For information, see “Generating lists in CATALOG MANAGER” on page 47.

2 Issue a command to invoke a utility. For a list of the commands that invoke specific utilities, see Table 45 on page 286 and Table 46 on page 287.

3 From a Utility List panel, in the Cmd column of a utility statement, type P.

4 Press Enter.

The Profile Selection panel is displayed (Figure 125).
Creating a utility profile from an existing profile

**Figure 125  Profile selection panel**

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Type</th>
<th>Profile ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Site</td>
<td>ALLDYNAMIC</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>ALLDYNAMICWITHGDGS</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>CHANGEMANY</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>DEFAULT</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>JCLCOPIES</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>PARTTEST</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>PEOPLESOFT</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>REGISTER TEST</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>RELOADOPTIONS</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>SHARELEVELCHANGE</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>TAPESTACKPARTS</td>
</tr>
<tr>
<td></td>
<td>Site</td>
<td>UNLOADOLYWITHFAILURANDOPTIONS</td>
</tr>
</tbody>
</table>

5 To select a profile, type **S** in the **Cmd** column beside the profile ID.

6 Press **Enter** to display the Utility List panel.

7 On a Utility List panel, type **ED** in the **Cmd** column.

8 Press **Enter**.

9 On the utility panel, make the applicable changes to the utility options values.

10 On the **Command** line, type **PROFILE SAVEAS profileID**, where **profileID** is the name of a new profile ID.

11 Press **Enter**.

   CATALOG MANAGER displays a message to confirm that your settings have been saved in the new profile ID.

12 Press **END** to display the Utility List panel.
Editing a utility profile

To change options values in a utility profile, complete the following steps:

1. Generate an object list.
   
   For information, see “Generating lists in CATALOG MANAGER” on page 47.

2. Issue a command to invoke a utility. For a list of the commands that invoke specific utilities, see Table 45 on page 286 and Table 46 on page 287.

3. From a Utility List panel, type P in the Cmd column of a utility statement.

4. Press Enter.

5. From the Profile selection panel, type S in the Cmd column beside the profile ID.

6. Press Enter to display the Utility List panel.

7. On a Utility List panel, type ED in the Cmd column.

8. Press Enter.

9. On the utility panel, make the applicable changes to the utility options values.

   If you edit only the first instance of a utility profile, most of the edits are propagated to the other instances of that utility profile. Propagation of the edits to subsequent rows is not displayed on the panel to allow faster processing. Edits that are not propagated or saved in utility profiles are those that are specific to DB2 objects, such as table names, column names, and partition numbers.

   To maximize performance, CATALOG MANAGER propagates utility profile edits down the object list. Therefore, if you want edits to apply to all objects in the utility list, edit the first object in the list.

10. On the Command line, type PROFILE SAVE.

11. Press Enter.

    CATALOG MANAGER displays a message to confirm that your changes have been saved.

12. Press END to display the Utility List panel.
Working with the last-used utility profile

CATALOG MANAGER includes a switch that displays, on the Utility List panel, the last-used utility profile ID for the selected utility type. If you often use the same utility profiles, CATALOG MANAGER automatically displays the last-used utility profile by default. If you do not want to see the last-used profile, you can remove it from the Utility List panel without resetting the switch. You can also display a list of available utility profiles and select a different one.

Setting the switch to display the last-used utility profile

1. Set the switch for the Last used prof field.
   
   For information, see “Setting CATALOG MANAGER switches” on page 95.

2. Press END to exit the panel.
   
   When the Utility List panel is displayed after you issue a utility command against an object, the last-used profile for the utility type is displayed in the Profile column.

Replacing the last-used profile with a different profile

To select a replacement for the last-used profile for the current utility job, complete the following steps:

1. From a Utility List panel, type P in the Cmd column of a utility statement.

2. Press Enter.

3. From the Profile selection panel, type S in the Cmd column beside the profile ID.

4. Press Enter to display the Utility List panel.
   
   The selected profile ID is displayed in the Profile column.

Removing the last-used profile from the utility list panel

To remove the last-used profile from the Utility List panel for the current utility job, type RP in the Cmd column, and then press Enter.

**NOTE**

The RP command is used only to remove the last-used profile. You cannot use the RP command to remove a profile ID that you have just selected.
Switching utility profiles

From any utility panel, you can switch from one utility profile to another without having to display the Utility List panel. You can use either of the following procedures.

**Selecting an alternative profile from a utility profile list**

To avoid typing the alternative profile name, complete the following steps:

1. On the **Command** line of a utility panel, type **PROFILE**.
2. To select a profile from the list of profiles for the selected utility type, type **S** in the **Cmd** column beside the profile ID.
3. Press **Enter**.

The utility panel is displayed and populated with the selected utility profile and its options values.

**Selecting an alternative profile by profile ID**

To avoid selecting the alternative profile from a list, on the **Command** line of a utility panel, type **PROFILE SET profileID**, and then press **Enter**. The utility panel is displayed and populated with the selected utility profile and its option values.

**TEMPLATE and LISTDEF control statements**

You can create **TEMPLATE** control statements to define the naming conventions and allocation of data sets, and you can create **LISTDEF** control statements to define lists of objects for utility processing.

Before using **LISTDEF** and **TEMPLATE** control statements in **CATALOG MANAGER**, you must create them in DB2. Then you must reference the data sets in the JCL Generation POF options. For information, see “**Setting the LISTDEF and TEMPLATE data set options**” on page 129.
Available utilities for TEMPLATE and LISTDEF statements

Table 49 lists the IBM utilities for which TEMPLATE and LISTDEF statements are valid.

<table>
<thead>
<tr>
<th>IBM utility</th>
<th>TEMPLATE statement valid?</th>
<th>LISTDEF statement valid?</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHECK DATA</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>CHECK INDEX</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CHECK LOB</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>COPY</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>COPYTOCOPY</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LOAD</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>MERGECOPY</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MODIFY RECOVERY</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MODIFY STATISTICS</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>QUIESCE</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>REBUILD INDEX</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>RECOVER</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>REORG INDEX</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>REORG TABLESPACE</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>REPORT</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>RUNSTATS</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Incorporating TEMPLATE and LISTDEF statements

CATALOG MANAGER includes TEMPLATE and LISTDEF selection fields on the panels for utilities for which the statements are applicable. To use the statements in utility jobs, complete the following steps:

1. Begin the utility job by issuing a DB2 utility command against an object.

   For a list of DB2 utility commands, see Table 46 on page 287.

2. On the Utility List panel, type **ED** in the **Cmd** column of the utility statement.

3. Press **Enter** to display the utility options panel.

4. Define values for the options as needed.
5 In the Listdef/Template Options field, type Y.

6 Press Enter.

The Listdef/Templates panel is displayed (Figure 126).

Figure 126 Listdef/Template options panel

```
Listdef/Templates: ACTADMN1.N1 1 to 5 of 5

Command ===>                                                  Scroll ===> PAGE
Listdef. . . . . . . .                                      Enter ? for list

Other Templates - Only DDs previously selected will be generated
PunchDDN template. . .                                      Enter ? for selection list
UNLDDN template. . . .                                      Enter ? for selection list
```

7 Select a LISTDEF statement name by using one of the following procedures:

- Type the desired LISTDEF statement name.
- Select from a list of LISTDEF statement names, as follows:
  
  A. In the Listdef field, type ? (question mark).
  
  B. Press Enter to display the selection list (Figure 127).
Incorporating TEMPLATE and LISTDEF statements

Chapter 9 Generating utility jobs

Figure 127 Listdef selection list panel

C. Type S beside the applicable LISTDEF name.

D. Press Enter to display the Listdef/Template Options panel. The Listdef field is populated with the selected LISTDEF statement name.

8 Select templates by using one of the following procedures:

- In the PunchDDN template or UNLDDN template field, type the desired statement name.

- Select from a list of statement names. Use a procedure similar to step 7.

9 Press END to display the utility panel.
The chapters listed in the following table discuss procedures that are available to administrators.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grant and revoke authorizations on catalog objects easily from the Primary Menu panel or an object list panel.</td>
<td>Chapter 10, “Managing authorizations”</td>
</tr>
<tr>
<td>Create and edit session profiles, which customize specific CATALOG MANAGER displays and operations for specific users or groups of users.</td>
<td>Chapter 11, “Customizing CATALOG MANAGER command access”</td>
</tr>
<tr>
<td>Browse and purge logs that CATALOG MANAGER maintains.</td>
<td>Chapter 12, “Maintaining logs”</td>
</tr>
</tbody>
</table>
Managing authorizations

This chapter contains the following topics:

Overview ................................................................. 305
Granting privileges by using the GRANT commands ...................... 306
  Granting privileges on a table ........................................ 307
  Granting privileges on a hierarchy of DB2 objects .................... 311
  Importing the SQL in another subsystem ........................... 314
  Copying an SQL_Table entry to another subsystem .................. 316
Granting privileges by issuing the COPYAUTHS command ................ 317
  Copying user ID privileges ........................................... 317
  Copying object privileges ............................................. 321
Granting privileges by generating SQL .................................... 321
Revoke privileges ..................................................... 322
  Preserving access to information ..................................... 323
  Generating the cascade report ....................................... 323
  Revoking privileges on specific objects ............................ 325
Where to go from here ................................................ 331

Overview

Managing authorizations within DB2 is one of the most important activities for a database or system administrator. DB2 provides a sophisticated system that enables you to control access to data resources at many levels. Due to the power and flexibility of this authorization system, the task of managing DB2 security can be complex.

CATALOG MANAGER makes it easy to find authorization information and to generate GRANT and REVOKE commands.
Granting privileges by using the GRANT commands

DB2 authorization information, except for that which applies to the INSTALL SYSADM, is maintained in tables in the DB2 catalog. CATALOG MANAGER supports the management of that information through familiar Interactive System Productivity Facility (ISPF) panels. As with other CATALOG MANAGER functions, you do not need to memorize Structured Query Language (SQL) statements or the names of the DB2 catalog tables or columns.

CATALOG MANAGER provides several tools to help you manage authorizations:

- The COPYAUTHS command enables you to copy privileges from one user ID to another user ID and from one object to another object easily, saving you the time and effort of issuing multiple GRANT commands.

- The Cascade Report shows you possible effects of a REVOKE action.

- The Reassign Grants option prevents you from losing authorizations when you execute a REVOKE by enabling you to assign those authorizations to another user ID.

- The Build SQLID before GRANT switch controls whether CATALOG MANAGER generates a SET CURRENT SQLID = grantor statement before each GRANT statement. For more information, see “Setting CATALOG MANAGER switches” on page 95.

Granting privileges by using the GRANT commands

CATALOG MANAGER does not provide capabilities to its users beyond the authorization that they already have, whether implicit or explicit. For example, if you have the authorization of a system administrator (SYSADM), you can perform system administrative actions through CATALOG MANAGER. If you have not been given an explicit authorization with the GRANT option enabled, you cannot use CATALOG MANAGER to grant that specific authorization to another user.

Because the object lists that CATALOG MANAGER produces make it so easy to review authorizations that have been granted, BMC recommends that you generate and see these lists to avoid unexpected results before you issue GRANT commands.

To view a list of the objects on which you can grant privileges, on the Command line, type CMD GRANT.
Granting privileges on a table

This procedure describes how to grant table privileges from the Table List panel. The privileges that you are able to grant depend on your authorization level.

1. Create a list of tables that includes the table or tables on which you want to grant privileges. For information, see "Generating lists in CATALOG MANAGER" on page 47.

2. In the Cmd (C) column beside the name of the table on which you want to grant privileges, type GRANT (GR) (Figure 128). You can specify any number of tables.

To grant privileges on every listed table, on the Command line, type GRANT ALL.

NOTE

The GRANT USAGE and GRANT DT commands display the Grant Distinct Type Privileges panel. The GRANT USE and GRANT BP commands display the Grant USE Privileges panel for buffer pools. While the GRANT USE command defaults to buffer pools, you can also use the following syntax for table spaces (GRANT USE ON TS) and storage groups (GRANT USE ON SG).

If a table list contains tables (T) and auxiliary tables (X), the GRANT ALL command excludes the auxiliary tables and the product issues a message. If the list contains only auxiliary tables (X), the GRANT ALL command allows only index authorizations to be granted. To ensure that the table list contains a specific type of table, issue the SEARCH TB command and specify the table type before you issue the GRANT ALL command on the table list.
Granting privileges on a table

### Figure 128  Table List panel with GRANT command

```
DEFF-R - ------------------------ TABLE LIST ------------------------ ROW 17 OF 1392
Command ===>
Scroll ===>

CMD will show commands for this list. Type command and press ENTER
Lists: AL CA CD CK CL CP C2 DB DS DT FK IC IM IS IX KC KU MK MQT MX NP OB
LIKE %,QZU%

<table>
<thead>
<tr>
<th>C</th>
<th>Table Name</th>
<th>Database</th>
<th>Tblspace</th>
<th>ColsPK</th>
<th>Type</th>
<th>Rows</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>QZU.QZU01_DACS03</td>
<td>QZUDAC</td>
<td>QZUS03AC</td>
<td>T</td>
<td>65K</td>
<td>4376</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>QZU.QZU01_DACS04</td>
<td>QZUDAC</td>
<td>QZUS04AC</td>
<td>T</td>
<td>65K</td>
<td>21K</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>QZU.QZU01_DACS05</td>
<td>QZUDAC</td>
<td>QZUS05AC</td>
<td>T</td>
<td>65K</td>
<td>21K</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>QZU.QZU01_DACS06</td>
<td>QZUDAC</td>
<td>QZUS06AC</td>
<td>T</td>
<td>65K</td>
<td>21K</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>QZU.QZU01_DACS07</td>
<td>QZUDAC</td>
<td>QZUS07AC</td>
<td>T</td>
<td>65K</td>
<td>21K</td>
<td></td>
</tr>
<tr>
<td>06</td>
<td>QZU.QZU01_DACS08</td>
<td>QZUDAC</td>
<td>QZUS08AC</td>
<td>T</td>
<td>65K</td>
<td>21K</td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>QZU.QZU01_DACS09</td>
<td>QZUDAC</td>
<td>QZUS09AC</td>
<td>T</td>
<td>975K</td>
<td>304K</td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>QZU.QZU01_DA1S01</td>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>T</td>
<td>2036</td>
<td>128</td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>QZU.QZU01_DA1S02</td>
<td>QZUDA1</td>
<td>QZUS02A1</td>
<td>T</td>
<td>2036</td>
<td>128</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>QZU.QZU01_DA1S03</td>
<td>QZUDA1</td>
<td>QZUS03A1</td>
<td>T</td>
<td>2036</td>
<td>129</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>QZU.QZU01_DA1S04</td>
<td>QZUDA1</td>
<td>QZUS04A1</td>
<td>T</td>
<td>2007</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>QZU.QZU01_DA1S05</td>
<td>QZUDA1</td>
<td>QZUS05A1</td>
<td>T</td>
<td>46K</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>QZU.QZU01_DB1S01</td>
<td>QZUDB1</td>
<td>QZUS01B1</td>
<td>T</td>
<td>120K</td>
<td>6150</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>QZU.QZU01_DB1S02</td>
<td>QZUDB1</td>
<td>QZUS02B1</td>
<td>T</td>
<td>120K</td>
<td>750</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>QZU.QZU01_DB1S03</td>
<td>QZUDB1</td>
<td>QZUS03B1</td>
<td>T</td>
<td>120K</td>
<td>3033</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>QZU.QZU01_DB1S04</td>
<td>QZUDB1</td>
<td>QZUS04B1</td>
<td>T</td>
<td>120K</td>
<td>1505</td>
<td></td>
</tr>
</tbody>
</table>
```

3 Press Enter.

The Grant Table Privileges panel is displayed (Figure 129).

### Figure 129  Grant Table Privileges panel

```
DEFF-R ----------- Grant Table Privileges ----------- 1 to 1 of 1
Command ===>
Scroll ===>

Generate SQL . N
Grant privileges to ...

AUTHIDs ... ...

ROLES  ...

With Grant ...

Table Privileges:
All ... N Index ... N Select ... N
Alter ... N Insert ... N Trigger ... N
Delete ... N References ... N Update ... N

Cmd  Creator.TBname

QZU.QZU01_DACS03

***************************************************************************** Bottom of data */
```
4 Specify the authorization IDs and the roles to which you want the privilege granted.

- To identify the grantees, in the AUTHIDs field, type a maximum of 10 authorization IDs.

- *(DB2 Version 9 or later)* To identify the roles, in the ROLES field, type a maximum of 5 roles.

5 In the With Grant field, type Y to enable the grantees to grant their table privileges to other users.

6 Specify which table privileges to grant, as follows:

- In the All field, type Y to grant all of the table privileges listed. If you used the GRANT ALL command on the object list panel, the default value in the All field is Y.

- In the Privileges fields, type Y beside the appropriate privileges to grant individual table privileges.

---

**NOTE**

Press HELP to display a brief description of the privileges.

---

7 After setting all of the panel values, in the Generate SQL field, type Y.

8 Press Enter.

The Confirm SQL panel is displayed *(Figure 130)*.
Figure 130  Confirm SQL panel for granting table privileges

9 On the Confirm SQL panel, you can edit and save the SQL and then execute it.

A  (optional) From the Command line, issue the SET sqlid command to change the value of the current SQLID.

--- NOTE ---

The ID shown in the Current SQLID field must have the proper authority to perform the specified SQL GRANT statement. If you hold a primary- or secondary-authorization ID that has the proper authority, you can change the Current SQLID to that authorization ID and complete the CREATE. To change the Current SQLID, use the SET command.

B  (optional) In the Edit options field, type Y to modify the default values for the options on the Confirm SQL panel. Then, press Enter.

The Options panel is displayed. In the Edit SQL and Confirm options field, type Y to display the options for the Confirm SQL panel. Press END to return to the Confirm SQL panel.

C  (optional) In the Edit SQL field, type Y to invoke an ISPF edit session to edit the SQL. Then, press Enter.

Press END to save the SQL and return to the Confirm SQL panel.
D (optional) In the **Save in SQL table** field, type A, Y, R, or N to specify whether to save the SQL in the CATALOG MANAGER SQL_Table.

<table>
<thead>
<tr>
<th>To perform this action</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>append the SQL to the SQL in the SQL_Table</td>
<td>A</td>
</tr>
<tr>
<td>save the SQL in the SQL_Table</td>
<td>Y</td>
</tr>
<tr>
<td>replace the SQL in the SQL_Table</td>
<td>R</td>
</tr>
<tr>
<td>discard the SQL</td>
<td>N</td>
</tr>
</tbody>
</table>

E (optional) In the **Name of saved SQL** field, type a name for the SQL.

F (optional) In the **Save in PDS** field, type Y to save the SQL in a member of a partitioned data set (PDS).

The saved SQL uses the ID displayed in the **Current SQLID** field as the object qualifier. If the SQL is not saved, the ID in the **Current SQLID** is used only to identify DB2 authority.

G (optional) In the **PDS(member)** field, type the name of the PDS and member.

H (optional) In the **Execute SQL** field, type Y to execute the SQL displayed on the Confirm SQL panel. Then, press Enter.

The SQL Progress Indicator panel is displayed. The panel automatically refreshes to display the status of the SQL that is being executed.

---

**Granting privileges on a hierarchy of DB2 objects**

The following procedure describes how to create the SQL to grant the privileges held by users on specified objects and their dependents.

1 Access the DB2 subsystem that contains the DB2 object hierarchy with authorizations.

2 Create a list of databases, table spaces, or tables. For information, see “Generating lists in CATALOG MANAGER” on page 47.
3 Enter the HGRANT command by using one of the following options:

- In the **Cmd** column of the list, type **HGRANT** beside the object that you want to use as a model. Then, press **Enter**.

- Alternatively, on the **Command** line, type the command and parameters with one space between them:
  
  1. **HGRANT**
  2. the object type (**DB**, **TS**, **TB**, or **VW**)
  3. the fully qualified name of an existing database, table space, table, or view
  4. **BATCH** (if you want to save the command for inclusion in a batch job)

  For example, type **HGRANT TS ACGRNT1.N1**.

  Then, press **Enter**.

  The SQL Progress Indicator panel is displayed. The indicator shows the execution of SQL that makes a list of dependents for the object specified.

  When SQL creation is complete, the Confirm SQL panel is displayed (Figure 131). This panel contains the generated GRANT statements for the source object and its dependents.
On the Confirm SQL panel, you can edit and save the SQL and then execute it:

A (optional) From the Command line, issue the SET sqlid command to change the value of the current SQLID.

NOTE
The ID shown in the Current SQLID field must have the proper authority to perform the specified SQL GRANT statement. If you hold a primary- or secondary-authorization ID that has the proper authority, you can change the Current SQLID to that authorization ID and complete the CREATE. To change the Current SQLID, use the SET command.

B (optional) In the Edit options field, type Y to modify the default values for the options on the Confirm SQL panel. Then, press Enter.

The Options panel is displayed. In the Edit SQL and Confirm options field, type Y to display the options for the Confirm SQL panel. Press END to return to the Confirm SQL panel.

C (optional) In the Edit SQL field, type Y to invoke an ISPF edit session to edit the SQL. Then, press Enter.

Press END to save the SQL and return to the Confirm SQL panel.
Importing the SQL in another subsystem

D (optional) In the **Save in SQL table** field, type A, Y, R, or N to specify whether to save the SQL in the CATALOG MANAGER SQL_Table.

<table>
<thead>
<tr>
<th>To perform this action</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>append the SQL to the SQL in the SQL_Table</td>
<td>A</td>
</tr>
<tr>
<td>save the SQL in the SQL_Table</td>
<td>Y</td>
</tr>
<tr>
<td>replace the SQL in the SQL_Table</td>
<td>R</td>
</tr>
<tr>
<td>discard the SQL</td>
<td>N</td>
</tr>
</tbody>
</table>

E (optional) In the **Name of saved SQL** field, type a name for the SQL.

F (optional) In the **Save in PDS** field, type Y to save the SQL in a member of a partitioned data set (PDS).

The saved SQL uses the ID displayed in the **Current SQLID** field as the object qualifier. If the SQL is not saved, the ID in the **Current SQLID** is used only to identify DB2 authority.

G (optional) In the **PDS(member)** field, type the name of the PDS and member.

To import the PDS member to another subsystem as an entry in the SQL_Table, see “Importing the SQL in another subsystem” on page 314.

H (optional) In the **Execute SQL** field, type Y to execute the SQL displayed on the Confirm SQL panel. Then, press Enter.

The SQL Progress Indicator panel is displayed. The panel automatically refreshes to display the status of the SQL that is being executed.

Importing the SQL in another subsystem

This procedure describes how to import the HGRANT SQL that you saved to a PDS member to another subsystem as an entry in the SQL_Table.

1 Complete the steps in “Granting privileges on a hierarchy of DB2 objects” on page 311. Save the SQL in a member of a PDS.

2 Press END to display the object list panel.

3 Use the CONNECT ssid command to access the target subsystem.

For information, see “Attaching to an SSID or server by using the connection selection list” on page 162.
4 On the target subsystem, on the **Command** line of the Primary Menu panel or an object list panel, type **IMPORT**.

5 Press **Enter**.

The Import SQL from a PDS panel is displayed (**Figure 132**).

**Figure 132 Import SQL from a PDS panel**

```
--------- Import SQL from a PDS ---------
Command ==>

Dataset imported from
Member pattern . . .            Member name or pattern for members to be copied
SQL Member owner . .  RDACRJ    Owner of member in SQL Table
SQL Member prefix               Member name prefix for SQL Table
Overwrite like names  N        (Y/N Y-overwrite members with the same name)
```

6 In the **Dataset imported from** field, type the data set name that includes the PDS member that contains the HGRANT SQL.

7 In the **Member pattern** field, type the PDS member name or a pattern that includes a wildcard.

8 (**optional**) In the **SQL Member owner** field, type a user ID to change the owner of the PDS when it is imported to the SQL_Table of the current subsystem.

9 (**optional**) In the **SQL Member prefix** field, type a prefix to attach to the PDS member name to change the name of the PDS member when it is imported to the SQL_Table of the current subsystem. The maximum number of characters for the prefix and member name is 18.

10 (**optional**) In the **Overwrite like names** field, type **Y** to replace a like-named SQL_Table entry on the current subsystem when the PDS is imported.

11 Press **Enter**.
If a match for the PDS member name or pattern is found, a selection list of member names is displayed.

12 Type S beside the names of all of the members that you want to import.

13 Press Enter.

The Import SQL from a PDS panel is displayed with a message that the selected members were copied.

14 To access and execute the HGRANT statements, press END to display the Primary Menu panel or object list panel.

**Copying an SQL_Table entry to another subsystem**

This procedure is useful if you have created a hierarchy of DB2 objects without authorizations, then moved those objects to another DB2 subsystem and want to grant the original authorizations on the moved objects.

1 On the Primary Menu panel or an object list panel, type SQL on the Command line.

2 Press Enter.

   The SQL Table List panel is displayed.

3 In the Cmd column, type CUT beside the SQL_Table entry that contains the GRANT statements.

4 Press Enter.

   The CATALOG MANAGER Clipboard is populated with the DCL.

5 Use the CONNECT ssid command to access the target subsystem. For information, see “Attaching to an SSID or server by using the connection selection list” on page 162.

6 On the Command line, type SQL.

7 Press Enter.

   The SQL Table List panel on the target subsystem is displayed.
Granting privileges by issuing the COPYAUTHS command

By using the COPYAUTHS command, you can duplicate the authorizations from one user ID to other user IDs, or from a specific DB2 object to other DB2 objects. This capability saves time and effort by enabling you to create a new DB2 object that has authorization requirements similar to an existing object—or provide a new user with authorizations similar to those of an existing user—without issuing multiple GRANT commands.

Copying user ID privileges

This procedure describes how to copy privileges from a source user ID to one or more target user IDs. You can copy all privileges or only specific privileges.

1 Create a list of user IDs that contains the source user ID, the one from which you want to copy authorizations.

For information, see “Generating lists in CATALOG MANAGER” on page 47.

2 In the User Authorizations List panel, type COPYAUTHS in the Cmd column beside the source user ID.

3 (optional) If the target user IDs are included on the list of user IDs, type the equal sign (=) in the Cmd column beside them.
4 Press Enter.

The Copy User Authorizations panel is displayed (Figure 133).

**Figure 133 Copy User Authorizations panel**

![Figure 133 Copy User Authorizations panel](image)

**NOTE**

The **Copy To** field identifies the user IDs that were selected as the targets for the authorizations in step 3. If you did not include target user IDs, then the **Copy To** field will be blank.

5 In the **Copy To** field, type the target user IDs. You can specify up to ten user IDs.

6 *(optional)* To identify a different source user ID, complete the following steps:

A In the **Copy From** field, type the new source user ID.

B In the **Copy To** field, type the target user IDs.
In the **Swap with Copy From** field, type Y beside the user ID that you now want to use as the source.

- If Y is indicated in the **Swap with Copy From** field for an object in the **Copy To** column, then that object becomes the source of the authorizations for all of the objects *including* the object in the **Copy From** field.

- If Y is indicated in the **Swap with Copy From** field for more than one object in the **Copy To** column, then only the *last* object marked as such becomes the source object. All other objects, including others marked with a Y, are treated as target objects.

7 In the **Include implicit privileges** field, type Y to grant implicit privileges from the source user ID to the target user IDs.

8 In the **Privilege type to include** field, type the code of the object type for which you want to copy authorizations.

9 In the **Generate copyauth grant SQL** field, type Y to generate SQL.

10 Press Enter.

The Confirm SQL panel is displayed (**Figure 134**).

**Figure 134  Confirm SQL panel for copying authorizations**

```
GRANT DBADM
ON DATABASE
QZUDSC30
TO RDACRJ ;
```

********** Bottom of data **********
On the Confirm SQL panel, you can edit and save the SQL and then execute it.

**A (optional)** From the Command line, issue the SET sqlid command to change the value of the current SQLID.

---

**NOTE**

The ID shown in the Current SQLID field must have the proper authority to perform the specified SQL GRANT statement. If you hold a primary- or secondary-authorization ID that has the proper authority, you can change the Current SQLID to that authorization ID and complete the CREATE. To change the Current SQLID, use the SET command.

**B (optional)** In the Edit options field, type Y to modify the default values for the options on the Confirm SQL panel. Then, press Enter.

The Options panel is displayed. In the Edit SQL and Confirm options field, type Y to display the options for the Confirm SQL panel. Press END to return to the Confirm SQL panel.

**C (optional)** In the Edit SQL field, type Y to invoke an ISPF edit session to edit the SQL. Then, press Enter.

Press END to save the SQL and return to the Confirm SQL panel.

**D (optional)** In the Save in SQL table field, type A, Y, R, or N to specify whether to save the SQL in the CATALOG MANAGER SQL_Table.

<table>
<thead>
<tr>
<th>To perform this action</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>append the SQL to the SQL in the SQL_Table</td>
<td>A</td>
</tr>
<tr>
<td>save the SQL in the SQL_Table</td>
<td>Y</td>
</tr>
<tr>
<td>replace the SQL in the SQL_Table</td>
<td>R</td>
</tr>
<tr>
<td>discard the SQL</td>
<td>N</td>
</tr>
</tbody>
</table>

**E (optional)** In the Name of saved SQL field, type a name for the SQL.

**F (optional)** In the Save in PDS field, type Y to save the SQL in a member of a partitioned data set (PDS).

The saved SQL uses the ID displayed in the Current SQLID field as the object qualifier. If the SQL is not saved, the ID in the Current SQLID is used only to identify DB2 authority.

**G (optional)** In the PDS(member) field, type the name of the PDS and member.

To import the PDS member to another subsystem as an entry in the SQL_Table, see “Importing the SQL in another subsystem” on page 314.
H (optional) In the Execute SQL field, type Y to execute the SQL displayed on the Confirm SQL panel. Then, press Enter.

The SQL Progress Indicator panel is displayed. The panel automatically refreshes to display the status of the SQL that is being executed.

Copying object privileges

The COPYAUTHS command also enables you to copy privileges that are held on one object to other objects of the same type. For example, you can copy the authorizations held on a table to another table, saving you the time of creating a GRANT statement for each user ID that has privileges on the source table. The procedure for copying privileges held on an object is similar to that for copying privileges from one user ID to another.

1 Create an object list that includes the source object. For information, see “Generating lists in CATALOG MANAGER” on page 47.

2 Follow the steps in “Copying user ID privileges” on page 317.

Granting privileges by generating SQL

You can generate SQL to re-create a set of privileges on an object or a set of privileges held by an authorization ID by issuing the DCL line command against the source object.

The BATCH parameter can be used with the DCL command. “Generating JCL for a job in batch” on page 206 explains the procedure for generating a job by using the DESCRIBE and HDESCRIBE commands. You can use a similar procedure by substituting the DCL command.

NOTE

The generated SQL includes only privileges that are explicitly granted.
Table 50 lists the valid source objects for the DCL command.

<table>
<thead>
<tr>
<th>Object code</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI</td>
<td>collection ID</td>
</tr>
<tr>
<td>DB</td>
<td>database</td>
</tr>
<tr>
<td>DT</td>
<td>data or distinct type</td>
</tr>
<tr>
<td>FN</td>
<td>function</td>
</tr>
<tr>
<td>PG</td>
<td>package</td>
</tr>
<tr>
<td>PL</td>
<td>plan</td>
</tr>
<tr>
<td>PR</td>
<td>procedure</td>
</tr>
<tr>
<td>SC</td>
<td>schema</td>
</tr>
<tr>
<td>SG</td>
<td>storage group</td>
</tr>
<tr>
<td>TB</td>
<td>table</td>
</tr>
<tr>
<td>TS</td>
<td>table space</td>
</tr>
<tr>
<td>UA</td>
<td>user authorization</td>
</tr>
<tr>
<td>US</td>
<td>user</td>
</tr>
<tr>
<td>VW</td>
<td>view</td>
</tr>
</tbody>
</table>

**Revoking privileges**

CATALOG MANAGER cannot be used to circumvent the DB2 rules for controlling access to data. Rules for revoking authorizations can be summarized as follows:

- Authorizations can be revoked only by a user ID with SYSADM or SYSCTRL authority, or by the user ID that granted the authorization.

- Authorizations that are granted implicitly cannot be revoked. For example, you cannot revoke the authorization to create a table from a user ID with DBADM authority because creating tables is a privilege that is implicitly available to DBADM authority.

- User IDs cannot revoke authorizations from themselves.

- No user ID (including those with SYSADM or SYSCTRL authority) can revoke privileges on a DB2 resource from the creator of the resource.

- Replicate authorizations that were granted to the same user ID from more than one grantor remain effective until revoked by each grantor.
Preserving access to information

The cascade effect of revoking privileges from a user ID requires administrators to pay careful attention to the consequences of issuing a REVOKE command. Access to information can be seriously interrupted in a complex authorization hierarchy if a REVOKE command is issued and the cumulative results have not been analyzed. CATALOG MANAGER makes it easy to access and review the possible implications of issuing a REVOKE command by providing the CASCADE command.

If the cascade effect of an intended REVOKE command is not acceptable, the Reassign GRANTOR capability enables you to retain the authorizations that would be revoked by assigning those authorizations to another SQLID.

Generating the cascade report

The cascade report enables you to view the hierarchy of authorizations for a DB2 object. This report is helpful for analyzing the impact of a REVOKE statement because it indicates any additional authorizations that are revoked if you execute the statement.

For example, if John Smith leaves your company, you will want to revoke his authorizations. However, Jane Jones’ authorizations might be based on Smith’s authorizations. By revoking Smith’s authorizations, you could be inadvertently revoking Jones’ authorizations. By generating and reviewing the cascade report, you can avoid unintended interruption of access to DB2 resources.

1 Create a system user list or a user list. For information, see “Generating lists in CATALOG MANAGER” on page 47.

2 In the system user list or user list, in the Cmd column, type CASCADE.

In the Cascade Report panel, the Report of Cascaded Grants section (Figure 135) shows which privileges are affected by the REVOKE action.
On the Cascade Report panel, you can edit and save the cascade report.

**A (optional)** From the Command line, issue the SET sqlid command to change the value of the current SQLID.

**NOTE**
The ID shown in the Current SQLID field must have the proper authority to perform the specified SQL statement. If you hold a primary- or secondary-authorization ID that has the proper authority, you can change the Current SQLID to that authorization ID and complete the CREATE. To change the Current SQLID, use the SET command.

**B (optional)** In the Edit options field, type Y to modify the default values for the options on the Cascade Report panel. Then, press Enter.

The Options panel is displayed. Press END to return to the Cascade Report panel.

**C** In the Edit the Cascade Report field, type Y to edit the report. Then, press Enter.

You edit the SQL in an ISPF edit session. Press END to save the SQL and return to the Cascade Report panel.

**D (optional)** In the Save in PDS field, type Y to save the report in a member of a partitioned data set (PDS).
**E** (optional) In the **PDS(member)** field, type the name of the PDS and member.

The saved report uses the ID displayed in the **Current SQLID** field as the object qualifier. If the SQL is not saved, the ID in the **Current SQLID** is used only to identify DB2 authority.

**F** (optional) In the **Save in SQL table** field, type **A**, **Y**, **R**, or **N** to specify whether to save the report in the CATALOG MANAGER SQL_Table.

<table>
<thead>
<tr>
<th>To perform this action</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>append the SQL to the SQL in the SQL_Table</td>
<td>A</td>
</tr>
<tr>
<td>save the SQL in the SQL_Table</td>
<td>Y</td>
</tr>
<tr>
<td>replace the SQL in the SQL_Table</td>
<td>R</td>
</tr>
<tr>
<td>discard the SQL</td>
<td>N</td>
</tr>
</tbody>
</table>

**G** (optional) In the **Name of saved SQL** field, type a name for the report.

4 Press END to display the object list panel.

**Revoking privileges on specific objects**

This procedure describes how to revoke user privileges on an object without losing cascaded privileges.

1 Generate an object list that contains the object from which you want to revoke user privileges. For information, see “Generating lists in CATALOG MANAGER” on page 47.

2 In the object list, in the **Cmd** column beside the source object, generate one of the following lists:

- To generate a list of the user privileges for the object, type **US**, and then press **Enter**. The example in **Figure 136** shows the user privileges for a database.
To generate a user list for that object, type `UA`, and then press `Enter` (Figure 137).

Figure 136  Object privileges panel

![Object privileges panel]

Figure 137  User List panel

![User List panel]
The columns on a object privileges panel or a User List panel reflect the privileges associated with the type of object from which the list was generated. To display more information about the fields on the panel, press HELP.

3 In the **Cmd** column beside the source user ID, type **REVOKE**.

4 Press **Enter**.

The Confirm SQL for Revoke Reassign panel is displayed (Figure 138).

**Figure 138  Confirm SQL for Revoke Reassign panel**

5 On the Confirm SQL for Revoke Reassign panel, you can perform several actions before executing the SQL to revoke privileges.

A From the **Command** line, issue the SET *sqlid* command to change the value of the current SQLID.

**NOTE**

The ID shown in the **Current SQLID** field must have the proper authority to perform the specified SQL REVOKE statement. If you hold a primary-or secondary-authorization ID that has the proper authority, you can change the **Current SQLID** to that authorization ID and complete the CREATE. To change the **Current SQLID**, use the SET command.
B In the **Reassign GRANTOR** field, type the SQLID of a new grantor for the cascading authorizations.

---

**NOTE**

With proper authorization, you can issue the SET *sqlid* command on the **Command** line of the User List panel to change the value of both the **Current SQLID** and the **Reassign GRANTOR** fields.

---

C In the **Edit options** field, type **Y** to modify the default values for the options on the Confirm SQL for Revoke Reassign panel. Then, press **Enter**.

The Options panel is displayed. In the **Edit SQL and Confirm options** field, type **Y** to display the options for the Confirm SQL panel. Press **END** to return to the Confirm SQL for Revoke Reassign panel.

D In the **Edit SQL** field, type **Y** to invoke an ISPF edit session to edit the SQL. Then, press **Enter**.

Press **END** to save the SQL and return to the Confirm SQL for Revoke Reassign panel.

E In the **Reassign Grants** field, type **Y** to reassign the privileges to the SQLID specified in the **Reassign GRANTOR** field. The product will generate executable GRANT statements with the SQLID specified in the **Reassign GRANTOR** field.

---

**WARNING**

Revoking a privilege that created a view also drops the view and any dependent INSTEAD OF triggers. To re-create the view and the triggers, you must specify **Y** for the **Reassign Grants** field. CATALOG MANAGER then generates the CREATE VIEW and CREATE TRIGGER statements and inserts them into the SQL after the REVOKE statement.

The CREATE VIEW statement still identifies the revoked SQLID as the creator of the view. Save and edit the SQL to change the creator.

The Confirm SQL for Revoke Reassign Grants panel is displayed (Figure 139).
Figure 139  Confirm SQL for Revoke Reassign Grants panel

The SQL section of the panel shows the sequence of the SQL execution. After revoking the user privileges, the new SQLID executes the GRANT statements.

In the View Cascade Report field, type Y to view the cascade report, which shows which privileges are affected when the REVOKE action is executed. Then, press Enter.

The Cascade List Report for Revoke/Reassign panel is displayed (Figure 140). For information about the Cascade List Report, see “Generating the cascade report” on page 323.
Revoking privileges on specific objects

Figure 140  Cascade List Report for Revoke/Reassign

G In the Save in SQL table field, type A, Y, R, or N to specify whether to save the SQL in the CATALOG MANAGER SQL_Table.

<table>
<thead>
<tr>
<th>To perform this action</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>append the SQL to the SQL in the SQL_Table</td>
<td>A</td>
</tr>
<tr>
<td>save the SQL in the SQL_Table</td>
<td>Y</td>
</tr>
<tr>
<td>replace the SQL in the SQL_Table</td>
<td>R</td>
</tr>
<tr>
<td>discard the SQL</td>
<td>N</td>
</tr>
</tbody>
</table>

H In the Name of saved SQL field, type a name for the SQL.

I In the Save in PDS field, type Y to save the SQL in a member of a partitioned data set (PDS).

The saved SQL uses the ID displayed in the Current SQLID field as the object qualifier. If the SQL is not saved, the ID in the Current SQLID is used only to identify DB2 authority.

J In the PDS(member) field, type the name of the PDS and member.
In the Execute SQL field, type Y to execute the SQL displayed on the Confirm SQL for Revoke Reassign panel. Then, press Enter.

The SQL Progress Indicator panel is displayed. The panel automatically refreshes to display the status of the SQL that is being executed.

**Where to go from here**

With CATALOG MANAGER, administrators can customize certain displays and features, either for convenience or control of access to data. This customization is possible from an individual or a sub-workgroup level.

Administrators and managers should read Chapter 11, “Customizing CATALOG MANAGER command access,” to learn how to use session profile features.
Customizing CATALOG MANAGER command access

This chapter contains the following topics:

Overview ................................................................. 333
Preparing to implement session profiles .................................. 334
Creating session profiles .................................................. 334
Commands to create and edit session profiles ............................ 335
Creating a session profile with a customized primary menu ........... 336
Displaying the session profiles list ......................................... 338
Displaying session profile descriptions .................................... 339
Creating a session profile with a tailored commands table ............ 340
Creating a session profile with an initial list filter ....................... 341
Editing session profiles .................................................... 342
Deleting session profiles .................................................. 342
Editing a customized Primary Menu ...................................... 343
Editing a tailored commands table ........................................ 344
Retrieving an initial list filter .............................................. 345
Assigning session profiles to users ......................................... 345
Determining a user’s capabilities .......................................... 345
Activating and deactivating session profiles .............................. 347
Where to go from here ..................................................... 348

Overview

With session profiles, administrators can customize specific CATALOG MANAGER displays and operations for specific users or groups of users. For example, administrators can allow DBAs to see only the CATALOG MANAGER items and functions the administrator specifies. Session profiles enable customization of the features shown in Table 51.
Preparing to implement session profiles

Administrators might use session profiles for the following reasons:

- to remove displayed menu items that certain users do not need or want to see
- to exclude commands that certain users should be restricted from executing
- to facilitate the retrieval of DB2 objects for certain users

You can include any or all of the custom features shown in Table 51 in a session profile.

### Table 51  Features supported by session profiles

<table>
<thead>
<tr>
<th>Feature</th>
<th>Customization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Menu panel</td>
<td>removes displayed actions and object types from the Primary Menu panel</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The functionality of the actions and object types that are not</td>
</tr>
<tr>
<td></td>
<td>displayed are still available to the user.</td>
</tr>
<tr>
<td>commands table</td>
<td>restricts usage of commands from the commands table for that session profile</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The user cannot issue commands that are excluded from the commands</td>
</tr>
<tr>
<td></td>
<td>table.</td>
</tr>
<tr>
<td>initial list filter</td>
<td>applies saved search variables by default when the user generates a list for</td>
</tr>
<tr>
<td></td>
<td>a specific object type</td>
</tr>
</tbody>
</table>

Preparing to implement session profiles

Any CATALOG MANAGER user can create session profiles, however, implementation of session profiles is available only to users with authorization to specify default option values, edit the CLIST, and issue the SET PROFILE `profileName` and SET PROFILE OFF commands.

Creating session profiles

You cannot create a session profile while you are working in session profile mode. You must first deactivate any session profile that might be active by issuing the SET PROFILE OFF command from the Command line of the Primary Menu panel or an object list panel.

Session profiles are saved in a DB2 table and can be created on each SSID to which CATALOG MANAGER is attached. Therefore, session profiles having the same name can be saved on different SSIDs.
Commands to create and edit session profiles

Use the commands that are shown in Table 52 to customize the Primary Menu and tailor the commands table when creating or editing a session profile.

Table 52  Session profile commands

<table>
<thead>
<tr>
<th>Command (short form)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOMIZE CU</td>
<td>displays the Menu Profile Customizing panel in which you customize the CATALOG MANAGER Primary Menu. This command can be entered in the following ways:</td>
</tr>
<tr>
<td></td>
<td>■ On the Command line, enter CUSTOMIZE profileName to create a new session profile or edit an existing session profile.</td>
</tr>
<tr>
<td></td>
<td>■ On the Profiles List panel, enter CU in the Cmd column to edit an existing session profile.</td>
</tr>
<tr>
<td>PROFILE PROFILES PRO</td>
<td>displays the Profiles List panel, which lists all of the saved profiles for CATALOG MANAGER. From the Profiles List panel, you can enter CU or TA in the Cmd column to edit a profile.</td>
</tr>
<tr>
<td>SET PROFILE</td>
<td>sets the current profile to the specified profile or resets it to default values. This command can be entered in the following ways:</td>
</tr>
<tr>
<td></td>
<td>■ Enter SET PROFILE profileName to set the current profile to a profile already saved</td>
</tr>
<tr>
<td></td>
<td>■ Enter SET PROFILE OFF to reset the profile to default values</td>
</tr>
<tr>
<td>TAILOR TA</td>
<td>displays the Profile Command Tailoring List panel in which you exclude commands from the CATALOG MANAGER commands table. This command can be entered in the following ways:</td>
</tr>
<tr>
<td></td>
<td>■ On the Command line, enter TAILOR profileName to create a new session profile or edit an existing session profile.</td>
</tr>
</tbody>
</table>
|                      |  ■ On the Profiles List panel, enter TA in the Cmd column to edit an existing session profile.
Creating a session profile with a customized primary menu

The following procedure describes how to create a session profile with a customized Primary Menu. To customize the Primary Menu, you remove actions and object types from the display.

**NOTE**

All actions and object types shown on the default Primary Menu are always available. Users can specify them by entering their codes even if they are removed from a customized menu.

1 On the Command line of the Primary Menu panel or an object list panel, type `CUSTOMIZE profileName`.

For this example, a session profile named LI_EXAMPLE is being created.

**NOTE**

The profile name can include a maximum of 18 characters.

2 Press Enter.

The Menu Profile Customizing panel is displayed (Figure 141).

**Figure 141** Menu Profile Customizing panel

```
DEFF-R ----------------- Menu Profile Customizing --------------------------
Command ==> Enter blank by items to exclude from the profile menu
 Y (L) List catalog objects (blank same as 0)
 Y (S) Search for catalog objects
 Y (C) Create objects
 Y (G) Grant privileges
 Y (O) CATALOG MANAGER options processing
 Y (D) DB2 Commands
 Y (M) Maintain logs menu
 Y (Q) List SQL for edit & execution
 Y (R) About this Release/ CATALOG MANAGER Quick Reference
 Y DB Database    Y TS Tablespace    Y PL Plan    Y ST Strings
 Y SG Stogroup    Y SU SysPrivUser  Y AL Alias    Y LO Location
 Y TB Table       Y SY Synonym      Y US User    Y CK Checks
 Y VM View        Y PG Package      Y CO Column  Y PR Procs
 Y IX Index       Y CJ Collection  Y DM DBRM
Preview the profile menu  N (Y/N)
Save the profile menu  N (Y/N)
Profile menu name is LI_EXAMPLE
Profile title is
```
3 Edit the menu by deleting the Y beside actions and object types that you want to remove from the menu.

4 In the Preview the profile menu field, type Y to review at the customized menu, and then press Enter.

The Preview Menu panel for the customized Primary Menu is displayed (Figure 142).

**Figure 142** Preview of customized Primary Menu panel

5 Examine the Preview Menu panel and note any desired changes.

6 Press Enter or END to display the Menu Profile Customizing panel.

7 (optional) In the Profile title is field, type the description of the profile. The profile title can include a maximum of 30 characters.

8 In the Save the profile menu field, type Y to save the customized menu.

9 Press Enter.

CATALOG MANAGER displays the Menu Profile Customizing panel with the confirmation message PROFILE. **profileName** ADDED.

10 Press END to exit the Menu Profile Customizing panel.
Perform the following steps to display the names of session profiles that have been created for the current SSID. You can check the list before creating a new session profile to ensure that you do not repeat a profile name.

1 Display a Primary Menu panel or an object list panel.

2 On the Command line, type PROFILE, and then press Enter.

The Profiles List panel is displayed (Figure 143).

![Profiles List panel](image)

For an explanation of the columns on the Profiles List panel, press HELP.

3 To display the Primary Menu panel or object list panel, press END.
Displaying session profile descriptions

The following procedure describes how to display the description of a session profile.

1. Display the Profiles List panel. For information, see “Displaying the session profiles list” on page 338.

2. In the Cmd column, type S beside the profile name that you want to view.

3. Press Enter.

The Describe Profile Entries panel is displayed (Figure 144).

Figure 144 Describe Profile Entries panel

The Describe Profile Entries panel displays the status of all types of CATALOG MANAGER customization, as follows:

- **Main menu** indicates whether the CATALOG MANAGER Primary Menu has been customized (Y or blank).
- **Command table** indicates whether the CATALOG MANAGER commands table has been tailored (Y or blank).
- The **object type** filter fields indicate whether a filter has been created for the object types shown (Y or blank).
Creating a session profile with a tailored commands table

The following procedure describes how to create a session profile with a tailored commands table. To tailor the commands table, you exclude commands from the default set of commands that is installed with CATALOG MANAGER.

NOTE
Commands that are excluded from the commands table are not available to the session profile user. CATALOG MANAGER returns the UNKNOWN COMMAND message if an unavailable command is attempted.

1 Use one of the following methods to display the Profile Command Tailoring List panel.

- To create a new session profile, type TAILOR profileName on the Command line of the Primary Menu panel or an object list panel, and then press Enter.

- To tailor the commands table for an existing session profile, type TA in the Cmd column of the session profile that you want to edit in the Profiles List panel (see “Displaying the session profiles list” on page 338), and then press Enter.

The Profile Command Tailoring List panel is displayed (Figure 145).
For an explanation of the columns on the Profile Command Tailoring List panel, press HELP.

2 To remove a command from the commands table, type X in the Exclude column beside the name of the command. You can scroll up and down to see the entire list of commands.

3 In the Save profile variables with commands currently excluded field, type Y to save the tailored commands table in the session profile.

4 Press Enter.

CATALOG MANAGER displays the Profile Command Tailoring List panel with the confirmation message PROFILEperfilName ADDED.

Creating a session profile with an initial list filter

The following procedure describes how to create an initial list filter. You can create one initial list filter for each object type to associate with a session profile. The initial list filter applies saved search variables by default when the user with that session profile generates a list for one of these object types.
Initial list filters are very similar to saved searches. For more detailed information about creating a saved search, see “Using SEARCH to generate lists based on object attributes” on page 183.

1 Generate a list of objects by using the SEARCH command. For more information, see “Generating a list by using the SEARCH command” on page 185.

2 Specify the variables under which to save the search in the Save Current Search Variables section of the Search Options panel:

A In the Owner field, type PROFILE.

B In the Name field, type the session profile name.

C Type an optional description in the Title field.

D In the Save current search variables using the following identification field, type Y.

E Press Enter.

CATALOG MANAGER displays the Search Options panel with the confirmation message PROFILE.profileNam ADDED.

Editing session profiles

After session profiles have been created and saved, the administrator might need to make changes to them. You can edit any or all of the three types of customization found in a session profile: the Primary Menu, the commands table, and initial list filters.

Deleting session profiles

Session profiles are stored in a DB2 table that is named BMCACTvr.Vvr_SEARCH_VARS (where vr represents the CATALOG MANAGER version number). You can easily delete a session profile by using the data editing feature of CATALOG MANAGER (see Chapter 6, “Browsing and editing data”), or you can write an interactive SQL statement to accomplish the same task.
Editing a customized Primary Menu

The following procedure describes how to edit the Primary Menu in a session profile that has already been created and saved.

**NOTE**

All actions and object types shown on the default Primary Menu are always available. Users can specify them by entering their codes even if they are removed from the customized menu.

1. On the Command line of the Primary Menu panel or an object list panel, type `CUSTOMIZE profileName`, and then press Enter.

   The Menu Profile Customizing panel for the named session profile is displayed (Figure 146).

**Figure 146** Menu Profile Customizing panel to edit a session profile

```
DEFF-R ----------------- Menu Profile Customizing --------------------------
Command ===> 

Enter blank by items to exclude from the profile menu

Y (L) List catalog objects (blank same as 0)
Y (S) Search for catalog objects
Y (C) Create objects
Y (G) Grant privileges
Y (O) CATALOG MANAGER options processing
Y (D) DB2 Commands
Y (M) Maintain logs menu
Y (Q) List SQL for edit & execution
Y (R) About this Release/CATALOG MANAGER Quick Reference

Y DB Database     Y TS Tablespace     Y PL Plan     Y ST Strings
Y SG Stogroup     Y SU SysPrivUser   Y AL Alias     Y LO Location
Y TB Table        Y SY Synonym       Y US User      Y CK Checks
Y VW View         Y PG Package       Y CO Column    Y PR Procs
Y IX Index        Y CI Collection    Y DM DBRM

Preview the profile menu N (Y/N)  
Save the profile menu N (Y/N)  
Profile menu name is SP_EXAMPLE  
Profile title is 
```

2. Edit the menu as needed.

   - Delete the Y beside actions and object types that you want to remove.
   - Type Y beside actions and object types that you want to restore to the menu.
Editing a tailored commands table

The following procedure describes how to edit the commands table that is already saved in a session profile.

1 On the Command line of the Primary Menu panel or an object list panel, type TAILOR profileName, and then press Enter.

   The Profile Command Tailoring panel for the named session profile is displayed.

2 Edit the commands table by typing X beside commands that you want to remove, or deleting X besides commands that you want to restore.

3 In the Save profile variables with commands currently excluded field, type Y to save the tailored commands table.

4 Press Enter.

   CATALOG MANAGER displays the Profile Command Tailoring List panel with the confirmation message PROFILE.profileName UPDATE.
Retrieving an initial list filter

To retrieve a list of initial list filters, follow the steps in “Generating a list by using the SEARCH command” on page 185. You can select a filter from the list to view, edit, or delete.

Assigning session profiles to users

After creating a session profile, the administrator assigns it to a specific user or group of users through settings in the BMCDB2 CLIST. The PR parameter in the BMCDB2 CLIST initially sets the session profile for all user groups. To turn off the session profile specified by the CLIST when using CATALOG MANAGER or to create another profile, you must have SYSADM authority, or the CRS installation option must be set to N. For more information about the BMCDB2 CLIST, see the installation guide.

You can further restrict access to CATALOG MANAGER functions through plan authorizations. For more information about plan authorizations, see Table 12 on page 69.

Determining a user’s capabilities

The following factors determine user capabilities (with regard to session profiles) that are specified in the installation options and CLIST:

- the value of the default option CRS, which governs authorization to issue the SET PROFILE profileName and SET PROFILE OFF commands

  The SET PROFILE profileName and SET PROFILE OFF commands activate and deactivate the named session profile during a CATALOG MANAGER session. These commands are issued by a specific user and apply to only that user during the current CATALOG MANAGER session.

- the BMCDB2 CLIST parameter PR, which determines which profile, if any, is invoked when a user starts CATALOG MANAGER
Table 53 describes how combinations of these factors determine user capabilities.

<table>
<thead>
<tr>
<th>CRS option</th>
<th>PR parameter</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRS=Y</td>
<td>profileName</td>
<td>When starting CATALOG MANAGER, all users on this subsystem access the session profile that is named in the PR parameter. Only users with SYSADM authority can issue the SET PROFILE ( \text{profileName} ) and SET PROFILE OFF commands.</td>
</tr>
<tr>
<td>blank</td>
<td></td>
<td>No session profile is specified in the PR parameter; therefore, no session profile is accessed by all users when CATALOG MANAGER is started. The default CATALOG MANAGER Primary Menu and commands table are available to all users, and no saved search variables are used by default for list generation. The absence of a profile name in the PR parameter means that no default session profile exists for all users. Administrators can create session profiles for specific users and assign those profiles through additional CLISTs. Only users with SYSADM authority can issue the SET PROFILE ( \text{profileName} ) and SET PROFILE OFF commands.</td>
</tr>
<tr>
<td>CRS=N</td>
<td>profileName</td>
<td>When starting CATALOG MANAGER, all users on this subsystem access the session profile that is named in the PR parameter. All users can issue the SET PROFILE ( \text{profileName} ) and SET PROFILE OFF commands.</td>
</tr>
<tr>
<td>blank</td>
<td></td>
<td>No session profile is specified in the PR parameter; therefore, no session profile is accessed by all users when CATALOG MANAGER is started. The default CATALOG MANAGER Primary Menu and commands table are available to all users, and no saved search variables are used by default for list generation. The absence of a profile name in the PR parameter means that no default session profile exists for all users. Administrators can create session profiles for specific users and assign those profiles through additional CLISTs. All users can issue the SET PROFILE ( \text{profileName} ) and SET PROFILE OFF commands.</td>
</tr>
</tbody>
</table>
Activating and deactivating session profiles

The following procedure describes how to make a session profile active or inactive for the current user during the current session. This procedure is used for testing session profiles.

**NOTE**
You must have authorization to issue the SET PROFILE `profileName` and SET PROFILE OFF commands to perform this procedure. For more information, see “Preparing to implement session profiles” on page 334.

1. Issue the SET PROFILE OFF command.

2. On the Command line of the Primary Menu panel or an object list panel, type `SET PROFILE profileName`, and then press Enter.

   The Primary Menu panel for the named session profile is displayed. Figure 147 shows the customized Primary Menu for the session profile named DATA_ENTRY.

   **NOTE**
   If no initial list filters have been associated with the active session profile, you cannot generate a list without a qualifier.
Where to go from here

**Figure 147 Customized Primary Menu panel**

![Customized Primary Menu panel diagram]

To deactivate the session profile, type `SET PROFILE OFF` on the Command line of the Primary Menu panel or an object list panel, and then press Enter.

CATALOG MANAGER displays the default Primary Menu panel.

Where to go from here

CATALOG MANAGER provides three logs that assist the database administrator by recording the actions of users while they are working with the DB2 catalog. Two of these logs can even be viewed and updated by other BMC products.

If you are a DB2 administrator, read Chapter 12, “Maintaining logs,” to learn about these helpful CATALOG MANAGER tools.
Maintaining logs

This chapter contains the following topics:

Overview ................................................................. 349
Accessing the logs ....................................................... 350
Using the Session Log .................................................. 351
  Browsing the Session Log ........................................... 351
  Purging the Session Log ........................................... 353
Using the DDL Audit Log ............................................. 355
Using the Drop Recovery Log ....................................... 356
  Browsing the Drop Recovery Log ................................. 356
  Purging the Drop Recovery Log ................................. 359
Where to go from here .................................................. 359

Overview

During the installation of CATALOG MANAGER, several DB2 tables are created to store various types of logged information. These tables, or logs, provide flexibility, efficiency, and integrity in reporting. User setup is minimal because logged information is captured automatically.

The logs that are maintained by CATALOG MANAGER are the Session Log, DDL Audit Log, and Drop Recovery Log. The log data is displayed in typical list panels. You can execute commands against items on the log list panels.

NOTE

To use the log maintenance functions described in this chapter, you must have the following authorities:

- EXECUTE authority on the CATALOG MANAGER Log Table Maintenance plan.
- To use functions to browse the logs, you must have SELECT authority.
- To use purge functions, you must have DELETE authority on the CATALOG MANAGER Log Table Maintenance plan.
Accessing the logs

To access the logs and log maintenance functions, on the Primary Menu panel or any list panel, on the Command line, type MAINTAIN (MAINT).

The Log Maintenance Menu panel is displayed (Figure 148).

**Figure 148  Log Maintenance Menu panel**

You can browse the entries in the logs and permanently purge the logs of old entries. These functions work similarly for all of the logs.

- In the Browse panel for each log, you can choose to display the entire log or you can enter qualifications that determine which log entries are displayed.

  The log is displayed in a list panel on which you issue the DESCRIBE command to see details of the entries.

- In the Purge panel for each log, enter a date and time to indicate which entries should be kept in the log.

  CATALOG MANAGER permanently removes all entries that are recorded before the date and time that you specify.
Using the Session Log

The Session Log captures information pertaining to actions invoked by users during a CATALOG MANAGER session. The installer determines which types of commands are captured by the Session Log.

Browsing the Session Log

1. On the Primary Menu panel or any list panel, on the Command line, type MAINTAIN (MAINT).


The Browse Session Log panel is displayed (Figure 149).

**Figure 149  Browse Session Log panel**

<table>
<thead>
<tr>
<th>Column</th>
<th>Operator</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>. . . =</td>
<td>yyyy-mm-dd-hh.mm.ss.nnnnnn</td>
</tr>
<tr>
<td>Authid</td>
<td>. . . . =</td>
<td></td>
</tr>
<tr>
<td>Session Id</td>
<td>. . . =</td>
<td></td>
</tr>
<tr>
<td>Return_Code</td>
<td>. . . =</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>. . . =</td>
<td></td>
</tr>
<tr>
<td>Object_Type</td>
<td>. . . =</td>
<td></td>
</tr>
<tr>
<td>Object_Qual</td>
<td>. . =</td>
<td></td>
</tr>
<tr>
<td>Object_Name</td>
<td>. . =</td>
<td></td>
</tr>
</tbody>
</table>

You may add additional conditions in the WHERE clause below

WHERE

The Browse Session Log panel enables you to indicate browse specifications—the combination of operators, values, and WHERE statements that generate a customized Session Log List. This list is a subset of the Session Log and includes only the data that you want to see.
### Browsing the Session Log

#### 3
On the Browse Session Log panel, in the **Authid** field, type and operator and a value for the authorization ID.

#### 4 *(optional)*
In the **Function** field, type a command or SQL action.

#### 5 *(optional)*
Specify a WHERE clause for the SQL statement.

If you include a WHERE clause, observe the following rules:

- Use the column names as they are displayed on the Browse Session Log panel.
- Use correct SQL case, punctuation, and syntax as required by DB2.
- Use only wildcards that are supported by DB2.

#### 6
Press Enter.

The Session Log List panel is displayed (Figure 150).

---

### Figure 150  Session Log List

<table>
<thead>
<tr>
<th>Command</th>
<th>Date</th>
<th>Authid</th>
<th>Sesn</th>
<th>Funcn</th>
<th>RtnC</th>
<th>TypName</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF-R</td>
<td>2010-09-20</td>
<td>RDATLF3</td>
<td>2</td>
<td>CREATE</td>
<td>000</td>
<td>TB RDATLF3.MXCDB06_TG01_.</td>
</tr>
<tr>
<td></td>
<td>2010-09-20</td>
<td>RDATLF3</td>
<td>2</td>
<td>COMMEN</td>
<td>000</td>
<td>TB RDATLF3.MXCDB06_TG01_.</td>
</tr>
<tr>
<td></td>
<td>2010-09-20</td>
<td>RDATLF3</td>
<td>2</td>
<td>LABEL</td>
<td>000</td>
<td>TB RDATLF3.MXCDB06_TG01_.</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.ATTR</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.ATTR_VAL</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.AUDIT_LOG</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.CRS_VAL</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.DLG</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.DLG_ATR</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.EDITOR_USERS</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.FLTRCMB_RSLT</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.GRID_MAPPING</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.MSG</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.RECOVERY_LOG</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.SEARCH_VARS</td>
</tr>
<tr>
<td></td>
<td>2010-05-26</td>
<td>RDATLF3</td>
<td>1</td>
<td>ALTER</td>
<td>000</td>
<td>TB ACT101.SESSION_LOG</td>
</tr>
</tbody>
</table>

---

### NOTE
If you do not indicate any specifications, CATALOG MANAGER attempts to display the entire Session Log.
7 In the **Cmd** column for the appropriate items, type **D** or **S** to view detailed information about entries on the list.

8 Press **Enter**.

The Describe Audit Log Entry panel is displayed (**Figure 151**), which shows information from the Session Log.

**Figure 151  Describe Audit Log Entry panel for a Session Log**

9 Press **END** to move to the next Describe Audit Log Entry panel (if you selected multiple items) or back to the Session Log List panel.

---

**Purging the Session Log**

Because the Session Log can record almost all of the commands issued by all CATALOG MANAGER users, it can quickly grow to an unmanageable size. The Session Log data sets can become full or grow into extents as you use CATALOG MANAGER each day. To release space, you can either purge the log to remove old entries permanently, or you can save old entries in archives.
1 On the Primary Menu panel or any list panel, on the **Command** line, type **MAINTAIN (MAINT)**.

2 On the Log Maintenance Menu panel, select **Purge Session Log**.

   The Purge Session Log panel is displayed (**Figure 152**).

**Figure 152  Purge Session Log panel**

3 In the **Timestamp** field, type an appropriate value, according to the format that is shown on the panel. The purge function will remove all entries that were recorded before the timestamp that you enter.

4 In the **Purge now** field, type **Y**.

   **WARNING**

   The purged entries cannot be retrieved.

5 Press Enter.

   After the entries are removed, CATALOG MANAGER displays the Purge Session Log panel with the message **PURGE COMPLETED**.
Using the DDL Audit Log

The DDL Audit Log captures and records the execution of all SQL and DSN commands that update the DB2 catalog. Audited events include the following commands:

- ALTER
- BIND
- COMMENT
- CREATE
- DROP
- FREE
- GRANT
- LABEL
- REBIND
- REVOKE
- SET
- START
- STOP

CATALOG MANAGER automatically maintains this mandatory log. The installer cannot customize the data that is written to the DDL Audit Log.

For information about browsing and purging the DDL Audit Log, see “Browsing the Session Log” on page 351 and “Purging the Session Log” on page 353.

Using the Drop Recovery Log

The Drop Recovery Log records all of the DDL necessary for CATALOG MANAGER to recover a dropped object structure, its dependents, and its data. The drop must have been initiated by one of the following BMC products:

- CATALOG MANAGER
- ALTER
- CHANGE MANAGER

**NOTE**

The Drop Recovery Log will not record the DDL to recover implicitly created databases, table spaces, tables, and indexes.
For more information about dropping and recovering objects, see Chapter 8, “Dropping and recovering objects.”

**Browsing the Drop Recovery Log**

1. On the Primary Menu panel or any list panel, on the **Command** line, type **MAINTAIN** (**MAINT**).

2. On the Log Maintenance Menu panel, select **Browse Drop Recovery Log**.

   The Browse Recovery Log panel is displayed (**Figure 153**).

   **Figure 153  Browse Recovery Log panel**

   ```
   DEFF-R ---------------------  Browse Recovery Log  ---------------------------
   Command ===> Type browse specifications and press Enter.
   Column Operator Value
   Timestamp . . . = yyyy-mm-dd-hh.mm.ss.nnnnnn
   Authid . . . . . =
   Object_Type . . =
   Object_Qual . . =
   Object_Name . . =
   Product . . . . =
   You may add additional conditions in the WHERE clause below
   WHERE
   ```

3. On the Browse Recovery Log panel, in the **Authid** field, type and operator and a value for the authorization ID.

4. *(optional)* Specify a **WHERE** clause for the SQL statement.

   If you include a **WHERE** clause, observe the following rules:

   - Use the column names as they are displayed on the Browse Session Log panel.
   - Use correct SQL case, punctuation, and syntax as required by DB2.
   - Use only wildcards that are supported by DB2.
5 Press Enter.

The Recovery Log List panel is displayed (Figure 154).

**Figure 154 Recovery Log List**

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Date</td>
<td>Time</td>
<td>AuthId</td>
<td>Product</td>
<td>Type</td>
<td>Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>------</td>
<td>--------</td>
<td>---------</td>
<td>------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2011-02-01 14.43 RDAPXB2 ACT010 TS DBXNAUT.SBXNCOL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2011-01-27 16.03 RDAPXB3 AEX010 TS PLBNAUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2011-01-19 16.25 RDAPXB2 ACT010 TS PLDBBA1.PLBLALT02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2011-01-19 14.09 RDAPXB2 ACT010 TS PLDBBA1.PLBLALTER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2011-01-17 14.16 RDAPXB2 ACT010 TS PXBQXPB.HASHPXB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2011-01-17 14.14 RDAPXB2 ACT010 TB PXBTMPDB.HASH_BY_RANGE_NOT_</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2011-01-07 15.52 RDAPXB2 AEX010 TS MXMLG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2011-01-07 15.46 RDAPXB2 AEX010 TS MXMLG5.MXSS01G5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2010-12-15 13.12 RDAPXB2 ACT010 TS QCHD14.PXBS0214</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2010-11-05 15.52 RDAPXB2 ACT010 DB PXBQXPB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2010-09-14 17.22 RDAPXB3 ACT010 TB QCH_LONG.QCH_LONG_NAME_TABL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2010-07-29 17.53 RDAPXB ACT010 TB ACTB.CORTAPST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2010-04-20 11.20 RDAPXB ACT010 TS ACTIBPX.BACTSPXB1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2010-04-20 11.09 RDAPXB ACT010 TS ACTQX13.ACTS0113</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2010-04-20 11.07 RDAPXB ACT010 TS ACTQX13.ACTS0113</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>2010-04-19 15.37 RDAPXB ACT010 TS ACTQV10.ACTSV010</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6 In the **Cmd** column for the appropriate items, type **L** to view detailed information about entries on the list.

7 Press Enter.

The Recovery Log Detail panel is displayed (Figure 155).
In the **Cmd** column for the appropriate items, type **D** or **S** to view detailed information about entries on the list.

**Press Enter.**

The Describe Audit Log Entry panel is displayed (Figure 156), which shows information from the Drop Recovery Log.
Purging the Drop Recovery Log

For information about purging the Drop Recovery Log, see “Purging the Session Log” on page 353.

Where to go from here

This chapter discussed how you can use the logs in CATALOG MANAGER to keep track of daily interactions with the DB2 catalog.

For detailed information about the commands, keywords, and symbolic variables that CATALOG MANAGER uses, see CATALOG MANAGER Help.
Generating JCL

This appendix contains the following topics:

The JCL Generation function .................................................. 361
  Modifying the ISPF skeletons and variables .............................. 362
  Sizing JCL Generation data sets ........................................... 363

The JCL Generation function

When you generate JCL for Execution, CATALOG MANAGER uses symbolic
variables to resolve all data set names that appear on the interface panels of the
components. Parameters (from the product options file, or POF, and Execution
panels) are passed to Batch Execution JCL Generation using the AJXIN or AJXPOFIN
input streams. These parameters include the names of input files, JCL files, and the
diagnostic output files.

After the data set names have been resolved, the JCL Generation component
performs the following functions:

- scans the input worklist for all utilities and commands that will require JCL
- analyzes each utility command for its DD requirements
Depending on the override options that you select, the following tasks might be performed for each identified DD:

— size the data sets

— use the values specified for using DASD or tape units, as well as the values for tape-related options

— use the values specified for the SMS and alternate SMS data sets, data set thresholds, and alternate units

— process the options for image copy GDGs

■ resolves the names of work data sets that JCL Generation passes from the JCL Generation option panels, such as SORTWORK, and the unload data sets that Execution uses

■ merges DD statements that are used by more than one command (for example, SYSUT1 or SORTWORK) to avoid duplicates and to use the highest estimated space

JCL Generation either performs standard ISPF file tailoring or simulates file tailoring by using compiled skeleton libraries (SLIBs) to generate JCL. A large number of symbolic variables are available to automatically vary the JCL generated according to, for example, the subsystem name and the database name.

The generated JCL includes DD statements for all data sets that the job or the Execution component needs, as well as the EXEC statement for the program and any necessary control parameters. For many of the work data sets that Execution uses, you can create the JCL for a cleanup job step that automatically deletes the work data sets at the end of the run. Other data sets are commented out in the JCL, providing you with the option to delete these as well.

### Modifying the ISPF skeletons and variables

The JCL Generation component uses standard ISPF file tailoring to generate all JCL. The $AJXDOC member in the HLQ.DBSLIB data set lists and briefly describes each non-DD and DD statement skeletons that JCL Generation uses. The member also lists the variables that JCL Generation uses to construct the default names for permanent data sets.

Each skeleton name in the $AJXDOC member has a corresponding member in the HLQ.DBSLIB data set. Some of the members that contain skeletons also provide comments, tips, and suggestions for using the skeletons.
You can specify up to five user-defined variables in the POF. The JCL Generation User Defined Variables panel allows you to specify variable names (up to eight characters). Each variable has a corresponding symbolic variable, as shown in Table 54. You can use the symbolic variables in your job cards or data set prefixes. For information about specifying the variables, see “Setting user variables” on page 130.

### Table 54 User-defined variables

<table>
<thead>
<tr>
<th>POF variable</th>
<th>Symbolic variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER_VAR1_CHAR</td>
<td>&amp;UVR1</td>
</tr>
<tr>
<td>USER_VAR2_CHAR</td>
<td>&amp;UVR2</td>
</tr>
<tr>
<td>USER_VAR3_CHAR</td>
<td>&amp;UVR3</td>
</tr>
<tr>
<td>USER_VAR4_CHAR</td>
<td>&amp;UVR4</td>
</tr>
<tr>
<td>USER_VAR5_CHAR</td>
<td>&amp;UVR5</td>
</tr>
</tbody>
</table>

Table 58 on page 390 and Table 59 on page 442 list and describe the symbolic and SLIB variables that JCL Generation uses and specifies the length of each variable.

To improve the performance of the JCL construction phase of JCL Generation, you can use an SLIB complier. However, if you edit SLIBs after installation, you must recompile them. For information, see Appendix B, “Using the Skeleton Library compiler.”

## Sizing JCL Generation data sets

You can use the JCL Generation data set sizing function to tailor the data set sizes when the JCL is built. The function gathers information from one of the following sources (shown in general order of accuracy, from most accurate to least accurate):

1. the BMC DASD MANAGER PLUS product statistics database (statistics that the BMCSTATS utility gathers)
2. the DB2 system catalog (statistics that the IBM RUNSTATS utility gathers)
3. the results of VSAM object sampling
4. the default data set allocation parameters that are set from the JCL Generation Individual Data Set Options panel

Whether it uses the BMCSTATS historical database or object sampling, or even if it does not use sizing, JCL Generation obtains some information from the DB2 system catalog.
The formulas for estimating data set size are taken from the documentation for the IBM DB2 utilities and from the documentation for the BMC utilities. Table 55 shows the statistics that JCL Generation uses for space estimation and the source of the statistics.

Table 55  Data set sizing values and sources

<table>
<thead>
<tr>
<th>Value</th>
<th>BMCSTATS</th>
<th>IBM RUNSTATS</th>
<th>VSAM sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of active pages</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>number of modified pages</td>
<td>X</td>
<td>NA</td>
<td>X</td>
</tr>
<tr>
<td>page size</td>
<td>NA</td>
<td>X</td>
<td>NA</td>
</tr>
<tr>
<td>maximum row length</td>
<td>NA</td>
<td>X</td>
<td>NA</td>
</tr>
<tr>
<td>average row length</td>
<td>X</td>
<td>NA</td>
<td>X</td>
</tr>
<tr>
<td>number of rows</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>number of non-clustering indexes</td>
<td>NA</td>
<td>X</td>
<td>NA</td>
</tr>
<tr>
<td>longest key</td>
<td>X</td>
<td>X</td>
<td>NA</td>
</tr>
<tr>
<td>number of foreign keys</td>
<td>NA</td>
<td>X</td>
<td>NA</td>
</tr>
<tr>
<td>number of indexes</td>
<td>X</td>
<td>X</td>
<td>NA</td>
</tr>
<tr>
<td>longest foreign key</td>
<td>NA</td>
<td>X</td>
<td>NA</td>
</tr>
</tbody>
</table>

_TIP_

To specify the data sizing method, see “Setting the JCL options for static data sets” on page 105.
Using the Skeleton Library compiler

This appendix includes the following topics:

- Overview ................................................................. 365
- Compiling SLIBs ......................................................... 366
- Changing SLIBs ....................................................... 367
  - Testing changes using ISPF file tailoring ....................... 367
  - Compiling changed SLIBs ....................................... 368
- Processing SLIBs ..................................................... 369
  - Generating the SLIB report .................................... 369

Overview

You can improve the performance of the JCL construction phase of JCL Generation by using the skeleton library (SLIB) compiler. The SLIB compiler is a tool that is supplied with the Administrative Products for DB2. The use of compiled SLIBs with JCL Generation’s runtime unit, which was designed to handle the compiled SLIBs, can eliminate the ISPF file tailoring process. Figure 157 illustrates the processing flow of the SLIB compiler.
This appendix describes the compiler, how to test SLIBs before compiling them, and the compiler’s associated runtime unit.

**Compiling SLIBs**

The SLIB compiler is written in REXX and runs as a batch TSO job. Each SLIB is compiled, assembled, and linked into a separately executable load module. The separation of load modules provides maximum flexibility and ease of use because no other dependencies between SLIBs or other object modules exist. You can change one SLIB without having to recompile or relink any other SLIBs or modules.
The SLIB compiler examines each line of the interpretive SLIB language and translates it into assembler source code with commands and instructions that directly interact with the runtime unit. The runtime unit processes the requests and builds the JCL. Like other compilers, the SLIB compiler translates a higher-level language into assembler instructions. Like compilers for C, C++, COBOL, and other languages, the SLIB compiler has its own runtime unit.

**NOTE**

The SLIB compiler does not support all options provided with ISPF file tailoring, but it does support all features that the Administrative products currently use.

The SLIB compiler assumes that the SLIB adheres to standard ISPF file tailoring rules and constructs. Because the compiler does little verification of SLIB syntax, BMC recommends that you verify changes by using standard ISPF file tailoring before you compile the SLIBs. For details, see “Testing changes using ISPF file tailoring” on page 367.

### Changing SLIBs

To change an SLIB, code the changes to your SLIB source. Before you compile the SLIB, you should use JCL Generation to test the changes by using ISPF standard file tailoring. Testing the changes ensures that the SLIB is coded correctly and that no ISPF-related errors exist.

You must recompile an SLIB each time that you change its source. Sample JCL for the SLIB compiler is in member AJXCOMPS in the HLQ.DBCNTL data set that BMC provides at installation. To customize the JCL to your shop’s standards, follow the directions provided in this member.

### Testing changes using ISPF file tailoring

The runtime unit attempts to process compiled SLIBs first. If it cannot process a compiled SLIB, it reverts to standard ISPF file tailoring. (If standard file tailoring is required, the SLIB that is being processed must exist in the ISPSLIB data set.)
Any of the following criteria force the runtime unit to use standard ISPF file tailoring for an SLIB:

- A compiled SLIB cannot be loaded from either STEPLIB or ISPLLIB.

  Remove or rename the compiled version of the SLIB in the STEPLIB or ISPLLIB. Removing the compiled version forces the runtime unit to process the uncompiled SLIB, using standard ISPF file tailoring.

- The first four bytes of the compiled SLIB contain binary zeros.

  Insert a new line 1 in the SLIB. Beginning in column 1, enter:

  \(\text{)CM NO-COMPILE}\)

  Then, compile the SLIB. The compiler recognizes this SLIB as noncompilable and builds a load module with binary zeros in the first four bytes. The binary zeros force the runtime unit to process this SLIB by using standard ISPF file tailoring.

- The ddname, \(\text{$USESTFT}\), is allocated as DUMMY to either the TSO session or the batch job that is being executed.

  Add the ddname \(\text{$USESTFT}\) to the JCL stream for batch jobs, or allocate it to your TSO session with the TSO ALLOC command. This option forces all SLIBs to be processed by standard ISPF file tailoring.

  For example, in Batch Execution JCL Generation, add the following JCL:

  \(\text{//$USESTFT DD DUMMY}\)

  The option remains in effect for the duration of the batch job. In foreground processing, issue the following command before entering the product:

  \(\text{TSO ALLOC FI($USESTFT) DA('NULLFILE') SHR}\)

  The option remains in effect until you log off or you issue the following command:

  \(\text{TSO FREE FI($USESTFT)}\)

### Compiling changed SLIBs

After you successfully test the SLIB using standard file tailoring, compile the SLIB into your production HLQ.UDBLINK library. Ensure that you remove the \(\text{)CM NO-COMPILE}\) option if you used it. Before you test the compiled SLIB, turn off any other options that you used, such as \(\text{$USESTFT}\).
BMC strongly recommends that you process all SLIBs as compiled SLIBs, because the runtime performance can be adversely affected by processing noncompiled SLIBs. You can use the report that the runtime unit generates to verify that you are running with compiled SLIBs and that the compile date on the SLIB is what you expect. For more information about the runtime report, see “Generating the SLIB report” on page 369.

## Processing SLIBs

The runtime unit replaces the ISPF file tailoring interface. In most cases, interfaces to the runtime unit are the same as interfaces to ISPF file tailoring. To improve runtime performance, the runtime unit offers special logic that is designed specifically for JCL Generation.

The runtime unit handles mixed mode processing of compiled and noncompiled SLIBs. However, mixed mode processing is not recommended because it compromises the improved performance that the runtime unit was designed to provide.

All JCL processing flows through the runtime unit. The runtime unit

- resolves all variables
- provides numeric data padding
- handles I/O
- processes standard file tailoring requests, when necessary
- builds a report about the JCL generation process

## Generating the SLIB report

The reporting feature of the runtime unit helps you determine which SLIBs were processed, how they were processed, and when they were last assembled. To use the SLIB reporting feature in Batch JCL Generation, perform the following steps:

1. Add the ddname JGENSRPT to your batch JCL stream in the step that executes AJXBMAIN with a DCB of the following parameters.

   \[
   \text{LRECL=80,RECFM=FB, BLKSIZE=6160,DSORG=PS}
   \]
2. Resubmit your job.

An example follows:

```sql
//JGENSRPT DD SYSOUT=*,
// DCB=(LRECL=80, BLKSIZE=6160, RECFM=FB, DSORG=PS)
```

If you need to produce this report in the foreground, you can use the TSO ALLOC command to allocate the ddname to any data set with RECFM=FB and LRECL=80. An example follows:

```sql
TSO ALLOC FI(JGENSRPT) DA('dataSet.name') SHR
```

In this example, `dataSet.name` is an existing sequential data set of RECFM=FB and LRECL=80.

**Figure 158** shows a sample runtime report.

**Figure 158  Sample runtime report (Part 1 of 2)**

<table>
<thead>
<tr>
<th>Skelname</th>
<th>Usage</th>
<th>Compile Date</th>
<th>Compile Time</th>
<th>Usage Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJX$ACMX</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.16</td>
<td>1</td>
</tr>
<tr>
<td>AJXJOB0</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.46</td>
<td>1</td>
</tr>
<tr>
<td>AJX#USRV</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.14</td>
<td>1</td>
</tr>
<tr>
<td>AJXJOB5</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.46</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTEP1</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.58</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTEP7</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.59</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTEPU</td>
<td>Compiled</td>
<td>08/28/2009</td>
<td>07.08</td>
<td>1</td>
</tr>
<tr>
<td>AJXSYSX$</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>06.02</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTWK0</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>06.00</td>
<td>1</td>
</tr>
<tr>
<td>AJXISPFM</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.45</td>
<td>1</td>
</tr>
<tr>
<td>AJXCLIBU</td>
<td>Compiled</td>
<td>09/26/2009</td>
<td>18.12</td>
<td>1</td>
</tr>
<tr>
<td>AJXMLIBU</td>
<td>Compiled</td>
<td>09/26/2009</td>
<td>18.12</td>
<td>2</td>
</tr>
<tr>
<td>AJXISPF5</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.45</td>
<td>1</td>
</tr>
<tr>
<td>AJXSLIBU</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.57</td>
<td>1</td>
</tr>
<tr>
<td>AJXTLIBU</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>06.02</td>
<td>1</td>
</tr>
<tr>
<td>AJXPLIBU</td>
<td>Compiled</td>
<td>09/26/2009</td>
<td>18.12</td>
<td>1</td>
</tr>
<tr>
<td>AJX#PRNT</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.12</td>
<td>1</td>
</tr>
<tr>
<td>AJXNOSTS</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.50</td>
<td>1</td>
</tr>
<tr>
<td>AJXWORK0</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>06.06</td>
<td>1</td>
</tr>
<tr>
<td>AJXWORK1</td>
<td>Compiled</td>
<td>10/11/2009</td>
<td>22.11</td>
<td>7</td>
</tr>
<tr>
<td>AJXWKUNT</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>06.06</td>
<td>7</td>
</tr>
<tr>
<td>AJX#MTAP</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.12</td>
<td>13</td>
</tr>
<tr>
<td>AJXESTIM</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.43</td>
<td>10</td>
</tr>
<tr>
<td>AJX#DSNS</td>
<td>Compiled</td>
<td>10/04/2009</td>
<td>05.10</td>
<td>9</td>
</tr>
</tbody>
</table>
The report summary at the end of Figure 158 provides the information shown in Table 56.

Table 56  Runtime report statistics

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of FTINCLs</td>
<td>the number of file tailoring FTINCL requests</td>
</tr>
<tr>
<td>Number of )IMs</td>
<td>the number of imbeds that are encountered when FTINCLs are processed</td>
</tr>
<tr>
<td>SLIBs processed</td>
<td>the number of SLIBs</td>
</tr>
<tr>
<td>Number of JCLRECs</td>
<td>the number of JCL records</td>
</tr>
<tr>
<td>Runtime units lastcc</td>
<td>the last condition code encountered</td>
</tr>
<tr>
<td>Runtime units maxrc</td>
<td>the highest return code encountered</td>
</tr>
</tbody>
</table>
Integrating CATALOG MANAGER for DB2 with SQL Explorer

This appendix contains the following topics:

Overview ................................................................. 373
Before you begin .................................................. 374
Commands to access SQL .......................................... 374
   Explaining a DBRM, package, or plan ...................... 375
   Explaining a statement from a DBRM or package ........ 376
   Explaining a statement from the SQL_Table ............... 379
   Editing the SQL statement by using the SQLX Edit Macro. .... 381

Overview

By integrating the BMC SQL Explorer product with CATALOG MANAGER, you can access and explain SQL in several ways. To access SQL statements in CATALOG MANAGER, you can create lists of DB2 objects and navigate through the objects to SQL statements that are stored in the DB2 catalog, or you can use the CATALOG MANAGER SQL_Table to access SQL statements that were saved in a CATALOG MANAGER session.

This section describes how to use CATALOG MANAGER commands to navigate to SQL Explorer. For detailed information about the functions of SQL Explorer and the reports that it produces, see the SQL Explorer for DB2 User Guide.
Before you begin

Before you attempt to use SQL Explorer with CATALOG MANAGER, verify that the ACTPSS CLIST has been installed and customized for SQL Explorer. For more information, see the *System and SQL Performance for DB2 Installation Guide*.

**Commands to access SQL**

CATALOG MANAGER enables you to issue simple line commands to access SQL, as follows:

- Issue the BMCEXPLORE command to explain a DBRM, package, or plan (Figure 159).

**Figure 159 Using the BMCEXPLORE command**

- Issue the DESCRIBE and GET commands to explain an SQL statement that is stored in a DBRM or a package (Figure 160).

**Figure 160 Using the DESCRIBE and GET commands**
- Issue the ANALYZE command to explain or edit a DML statement from the CATALOG MANAGER SQL_Table (Figure 161).

**Figure 161 Using the ANALYZE command**

![Diagram showing the process of using the ANALYZE command to explain or edit a DML statement.]

### Explaining a DBRM, package, or plan

In this task, you use the BMCEXPLORE command to access SQL Explorer to explain DBRMs, packages, and plans.

1. Generate a list of eligible objects. For information, see “Generating lists in CATALOG MANAGER” on page 47.

2. In the Command (Cm) column beside the source object, type BMCEXPLORE (BMCEX) (as shown in Figure 162).
Explaining a statement from a DBRM or package

In this task, you use the DESCRIBE command to access SQL Explorer to explain individual SQL statements in DBRMs and packages.

The following types of statements are valid:

- SELECT
- INSERT
- UPDATE
- DELETE
- DECLARE CURSOR (static SQL only)

1. Generate a list of eligible objects. For information, see “Generating lists in CATALOG MANAGER” on page 47.

2. In the Command (Cm) column beside the source object, type DESCRIBE.

3. Press Enter to begin SQL Explorer Explain processing. For more information, see the SQL Explorer for DB2 User Guide.
3 Press Enter.

The DESCRIBE panel for the source object is displayed (Figure 163).

**Figure 163  DESCRIBE panel**

<table>
<thead>
<tr>
<th>Stmtno</th>
<th>Stmt</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>DECLARE  CRS1 CURSOR FOR</td>
</tr>
<tr>
<td></td>
<td>SELECT COLUMN_1</td>
</tr>
<tr>
<td></td>
<td>FROM QZU.QZUT01_D15S01</td>
</tr>
<tr>
<td></td>
<td>WHERE COLUMN_1 = 1234</td>
</tr>
<tr>
<td>29</td>
<td>OPEN CRS1</td>
</tr>
<tr>
<td>34</td>
<td>FETCH CRS1</td>
</tr>
<tr>
<td></td>
<td>INTO :iColumn1</td>
</tr>
<tr>
<td>38</td>
<td>CLOSE CRS1</td>
</tr>
</tbody>
</table>

4 Scroll down the DESCRIBE panel to locate the statement.

5 On the Command line, type GET nnn, where nnn is the number of the statement to be explained (Figure 164).
Explaining a statement from a DBRM or package

**Figure 164 Use of GET subcommand**

```
DEFF-R   Package: DSN_DEFAULT_COLLID_QZUTSTPL.QZUTSTPL   Line 22 of 42 Col 1 80
Command ===> GET 24  Scroll ===> PAGE

Reoptvar . . . N
Keepdynamic . . N
Type . . . . DBprotocol . . D
Functions . . . . 15-17.15.57.041490 Opthint . . .
Encoding CCSID . 37
Relbound . . . O
Remarks . . . .
Rounding . . . E
Lastused . . . 01/01/0001

Statement Stmt
24    DECLARE CRS1 CURSOR FOR
       SELECT COLUMN_1
       FROM QZU.QZUT01_D15S01
       WHERE COLUMN_1 = 1234
29    OPEN CRS1
34    FETCH CRS1
38    CLOSE CRS1

END OF DATA
```

**6 Press Enter.**

The Confirm SQL panel is displayed (Figure 165).

**Figure 165 Confirm SQL panel**

```
DEFF-R   Confirm SQL   Line 22 of 42 Col 1 80
Command ===>  Scroll ===> PAGE

Current SQLID . . . . . RDACRJ
Edit Options . . . . . N   Y/N Modify SQL processing options
Edit SQL . . . . . . . . N   Y/N Edit SQL before executing
Save in SQL table . . . N   A/Y/R/N A/Y-Append, R-Replace
Name of saved SQL . . . 20110121_115552
Save in PDS . . . . . . . N   Y/N Save SQL in PDS
PDS(member) . . . . . .
Analysis . . . . . . . . N   Y/N Call SQL Explorer for EXPLAIN
Edit/Browse data . . . . N   E/B/N Call the Table Editor

Execute SQL . . . . . . . N   Y/N Execute the SQL

SELECT COLUMN_1
FROM QZU.QZUT01_D15S01
WHERE COLUMN_1 = 1234

******************************* Bottom of data *******************************
```
Explaining a statement from the SQL_Table

7 To continue, choose one of the following procedures:

- To edit the SQL statement before explaining it, or to explain or execute the SQL statement on another subsystem, see “Editing the SQL statement by using the SQLX Edit Macro” on page 381.

- To begin SQL Explorer Explain processing, in the Analysis field, enter Y.

For more information, see the SQL Explorer for DB2 User Guide.

Explaining a statement from the SQL_Table

In this task, you access SQL Explorer to explain SQL statements that have been saved in the CATALOG MANAGER SQL_Table.

The following types of statements are valid:

- SELECT
- INSERT
- UPDATE
- DELETE
- DECLARE CURSOR (static SQL only)

1 Display the CATALOG MANAGER SQL_Table List panel. For more information, see “Displaying the SQL_Table list” on page 213.

2 In the Command (Cmd) column beside the source statement, type ANALYZE (Figure 166).
Explaining a statement from the SQL_Table

Figure 166  SQL_Table List panel

3  Press Enter.

The Confirm SQL panel is displayed (Figure 167).

Figure 167  Confirm SQL panel
To continue, choose one of the following procedures:

- To edit the SQL statement before explaining it, or to explain or execute the SQL statement on another subsystem, see “Editing the SQL statement by using the SQLX Edit Macro” on page 381.

- To begin SQL Explorer Explain processing, in the Analysis field, enter Y.

For more information, see the SQL Explorer for DB2 User Guide.

**Editing the SQL statement by using the SQLX Edit Macro**

1. Explain a statement from the SQL_Table. For information, see “Explaining a statement from the SQL_Table” on page 379.

2. On the Confirm SQL panel, in the Edit SQL field, type Y.

3. Press Enter.

An ISPF edit panel is displayed (Figure 168).

**Figure 168  ISPF Edit panel**

```
EDIT       RDCARJ.BMCCAT.WORK                              Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
****** ***************************** Top of Data *****************************
==MSG> -Warning- The UNDO command is not available until you change
your edit profile using the command RECOVERY ON.
000001 SELECT *
000002 FROM QCHDBIL2.QCHSP03_TT31B
000003 ;
000004
000005 INSERT INTO QCHDBIL2.QCHSP03_TT31B (  
000006   SSSNO,
000007   EMPNO,
000008   EMPNAME,
000009   EMPDEPT,
000010   EMPDEPTNAME,
000011   EMPHIREDATE,
000012   EMPDOB,
000013   EMPAGE,
000014   EMSALARY,
000015   EMPSEX
000016 ) VALUES (  
000017   ‘11’,                  --CHAR(9) SSSNO
```

4. Edit the SQL statement as necessary.
5 Mark the SQL text that you want to analyze by entering the block QQ command at the statement begin line and at the statement end line.

--- NOTE ---
SQLX does not support the use of a sequential text file that is greater than 80 bytes.

Depending on the programming language, the begin and end statements may have different keywords.

--- NOTE ---
If the text spans multiple panels, you might get an INVALID COMMAND message when you press DOWN because ISPF does not recognize QQ as a valid command. Ignore the INVALID COMMAND message while you continue to mark the statement, then proceed to step 6 to initiate the macro. Alternatively, you can use the Qnn command at the beginning of the statement, where nn is the number of lines to search to locate the end of the statement.

6 On the Command line, type SQLX ssid, where ssid is the ID of the subsystem on which you want to Explain the SQL statement.

7 Press Enter to begin SQL Explorer Explain processing. For more information, see the SQL Explorer for DB2 User Guide.
Keywords and variables

This appendix contains the following topics:

Keywords ................................................................. 383
  AEXIN keywords .................................................. 383
  POF keywords ....................................................... 390
Symbolic variables ...................................................... 441

Keywords

Several option values control the operating default environment. These values are defined in the installation options module, an assembler module that CATALOG MANAGER uses.

AEXIN keywords

Table 57 lists the keywords in the AEXIN input stream.

Table 57  AEXIN keywords  (Part 1 of 7)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2MEGSQL</td>
<td>instructs the Execution component to allocate a 2-MB buffer for large SQL statements</td>
</tr>
<tr>
<td>ACM</td>
<td>specifies the BMC CHANGE MANAGER product</td>
</tr>
<tr>
<td>ALTERID</td>
<td>specifies the name of the ALTER ID</td>
</tr>
<tr>
<td>ALU</td>
<td>specifies the BMC ALTER product</td>
</tr>
<tr>
<td>ASU</td>
<td>specifies the BMC DASD MANAGER PLUS product</td>
</tr>
<tr>
<td>AUC</td>
<td>specifies the CM/PILOT component of the BMC CHANGE MANAGER product</td>
</tr>
</tbody>
</table>
## Table 57  AEXIN keywords (Part 2 of 7)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
</table>
| BINDFAIL        | causes worklist execution to stop with a return code of 8 if a bind fails.  
The halt will be noted in the sync tables, and an Execution restart will continue with the command that caused the failure.  
Without this parameter, worklist execution continues if a bind fails. |
| CATAUDIT        | if the CATAUDIT installation option is set to N and you manually add the keyword to the AEXIN input stream, instructs Execution to invoke the BMC CATALOG MANAGER product, override the installation option, and log executed DDL statements in the CATALOG MANAGER DDL Audit Log.  
If the keyword is not in the AEXIN input stream, Execution reads the ALTER or CHANGE MANAGER installation options. If CATAUDIT=Y, Execution audits the worklist.  
JCL Generation includes the CATALOG MANAGER installation option in the AEXIN input stream. The name of the installation option is passed from the BMCDB2 control table to CATALOG MANAGER when the product is invoked.  
This parameter is useful only if you have installed CATALOG MANAGER. You must use CATALOG MANAGER to recover any dropped objects. |
| CATDOPT name    | provides the ALTER or CHANGE MANAGER products with the name of the installation options module for the BMC CATALOG MANAGER product                                                                                                                                                                                                                       |
| CATRECOVER      | if the CATRECOV installation option is set to N and the keyword is in the AEXIN input stream, instructs Execution to invoke the BMC CATALOG MANAGER product and to log the information that is required to recover any objects that are dropped in the worklist.  
You can manually insert the keyword into the AEXIN input stream, or you can use the Drop Recovery option on the Execution Override Options panel.  
JCL Generation reads the ALTER or CHANGE MANAGER installation options. If CATRECOV=Y, JCL Generation inserts the keyword into the AEXIN input stream. If you set the CATRECOV installation option to N and rerun Execution, or if you manually remove the keyword from the AEXIN input stream, the dropped objects are not logged or rebuilt.  
JCL Generation includes the CATALOG MANAGER installation option in the AEXIN input stream. The name of the installation option is passed from the BMCDB2 control table to CATALOG MANAGER when the product is invoked.  
This parameter is useful only if you have installed CATALOG MANAGER. You must use CATALOG MANAGER to recover any dropped objects. |
<p>| CATUTIL         | specifies the worklist job                                                                                                                                                                                                                                                                                                                                    |</p>
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
</table>
| CHECKOPT        | provides the BMC products with the name of the options module for the BMC CHECK PLUS product  
The BMC_CHECK_OPTS keyword in the AJXPOFIN input stream replaces this keyword. |
| COPYOPT         | provides the BMC products with the name of the options module for the BMC COPY PLUS product  
The BMC_COPY_OPTS keyword in the AJXPOFIN input stream replaces this keyword. |
| DASDDOPT name   | provides ALTER or CHANGE MANAGER with the name of the installation options module for the BMC DASD MANAGER PLUS product  
This keyword allows Execution to access the BMCSTATS utility. |
| DB2STMSGS       | instructs Execution to send to AEXPRINT all of the messages that are returned from DB2 when a DB2 STOP command is issued  
Usually, these messages are suppressed, and the contents are analyzed by the STOPWAIT processor. However, if the stop does not occur, you might want to view the original messages. |
| DEBUGUNLD       | enables debugging user exits, if the exits are coded to use the keyword  
This keyword also instructs Execution to turn on bit VAUNFDBG in flag field VAUNFLAG. Execution passes this bit to the user exit in the control block. |
| DYNWORKUNIT     | for the Database Administration solution, defines the unit (such as SYSDA) that Execution uses to dynamically allocate temporary work data sets |
| ENV             | instructs the component to print ALTER or CHANGE MANAGER environment information (including a list of indexes Analysis uses) in its diagnostic output  
The ALUIN input stream also uses this keyword. |
| FLOW            | causes Execution to produce flow trace messages in AEXPRINT when entering and exiting modules |
| HASHFAIL        | causes Execution to terminate the job if a hash failure, such as a changed or added statement, occurs in a worklist |
| HASHWARNRC returnCode | defines the return code (returnCode) that Execution sends back when it finds only hash warnings  
Do not use 8 for this value. |
ITERATIONMODE enables Execution to run a REXX executable that generates utility syntax or SQL for Execution to run based on user-defined criteria. This utility syntax or SQL must be in a worklist format. You are responsible for creating the JCL that would enable the utility to run correctly (such as using dynamically allocated work files in the utilities). The REXX executable runs iteratively until it returns a code greater than 4 to Execution. You are responsible for restarting any commands that failed. Execution does not record any actions in the sync table. You must manually insert this keyword in the AEXIN input stream.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINES n</td>
<td>provides Execution with the number (n) of output lines per print page for AEXPRINT. The default is 56 lines per page. If you specify this keyword, you might need to change or add a /*JOBPARM statement in your JCL. You can manually add or modify this keyword.</td>
</tr>
<tr>
<td>LOADOPT</td>
<td>provides the BMC products with the name of the options module for the BMC LOADPLUS product The BMC_LOAD_OPTS keyword in the AJXPOFIN input stream replaces this keyword.</td>
</tr>
<tr>
<td>NOAPFOK</td>
<td>does not perform APF checking</td>
</tr>
</tbody>
</table>
| NOFAILNOIMAGECPY | instructs Execution to invoke the Drop Recovery feature of the BMC CATALOG MANAGER product and to allow an object to be dropped when the following conditions exist:  
  - No image copies of the object exist.  
  - The CATRECOVER keyword is specified in the AEXIN input stream.  
  The product automatically generates the NOFAILNOIMAGECPY keyword in the AEXIN input stream when one of the following conditions exists:  
  - The DROPR_NOIC POF keyword is set to Y.  
  - The NOFAILNOIMAGECPY option is set to Y on the JCL Generation Debugging, Display and Execution panel.  
  This keyword is useful only if you have installed CATALOG MANAGER.  
  This keyword overrides the DROPR_NOIC keyword in the AJXPOFIN input stream. |
| NOLOADCOMP       | instructs Execution not to compress extra spaces out of LOAD statements     |
| NOSQLCOMP        | instructs Execution not to compress extra spaces out of SQL statements      |
| NOSTARTOVER      | instructs Execution not to start the worklist again from the beginning of the worklist  
  See also the STARTOVER keyword in this table. |
### Table 57  AEXIN keywords (Part 5 of 7)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTIFYUNLD ( n )</td>
<td>instructs Execution to send a message to AEXPRINT every ( n ) records during an unload</td>
</tr>
<tr>
<td>NOWKIDREPLACE</td>
<td>instructs Import not to replace the changes in an existing work ID with an imported file</td>
</tr>
<tr>
<td>REBINDFAIL</td>
<td>causes worklist execution to stop with return code 8 if a rebind fails. The stop is noted in the sync tables, and an Execution restart continues with the command that caused the failure. Without this parameter, worklist execution continues if a rebind fails.</td>
</tr>
<tr>
<td>REBINDRC ( n )</td>
<td>allows worklist execution to continue if a rebind fails, but returns the ( n ) value for a final condition code instead of 4, the default value. When running standard JCL, the condition code is added to the step subsequent to the REBIND step. Execution writes warning messages to AEXPRINT but does not post entries in the sync tables.</td>
</tr>
<tr>
<td>RECOVEROPT</td>
<td>provides the BMC products with the name of the options module for the BMC RECOVER PLUS product. The BMC_RECOVER_OPTS keyword in the AJXPOFIN input stream replaces this keyword.</td>
</tr>
<tr>
<td>REORGOPT</td>
<td>provides the BMC products with the name of the options module for the BMC REORG PLUS product. The BMC_REORG_OPTS keyword in the AJXPOFIN input stream replaces this keyword.</td>
</tr>
<tr>
<td>RESTART</td>
<td>instructs Execution to restart a worklist from the last sync or stop point. RESTART fails if no -STOP command or error sync point (-SYNC) exists in the worklist. You cannot specify the RESTART keyword with the STARTOVER keyword.</td>
</tr>
<tr>
<td>RESTARTParm ( parameter )</td>
<td>during Execution restart, passes a user-defined parameter string ( (parameter) ) to the utility that is being restarted. The form of the parameter string depends on the utility that is being restarted.</td>
</tr>
<tr>
<td>SPBXPRINT</td>
<td>prints the output from a stored procedure</td>
</tr>
</tbody>
</table>
### SSID ssid

Identifies the DB2 subsystem ID or the DB2 data sharing group attachment name.

The SSID keyword must match the -SSID command in the worklist. This parameter is required.

If the SSID is specified in the JCL in an EXEC statement in a cataloged procedure (which begins with a PROC statement), its value overrides the value of the SSID keyword.

The AJXIN and ALUIN input streams also use this keyword.

### STARTOVER

Instructs Execution to start the worklist again from the beginning of the worklist.

You cannot specify the STARTOVER keyword with the RESTART keyword. See also the NOSTARTOVER keyword in this table.

### STATS

Prints the execution statistics.

### STOPWAIT n

Specifies the number (n) of intervals to wait for a DB2 STOP command to stop a database or table space.

The first interval is 10 seconds, and the second through tenth intervals are an additional 30 seconds each. Examples are as follows:

- \( n = 1 \), total wait = 10 seconds
- \( n = 2 \), total wait = 40 seconds
- \( n = 3 \), total wait = 70 seconds
- \( n = 4 \), total wait = 100 seconds
- \( n = 5 \), total wait = 130 seconds
- \( n = 6 \), total wait = 160 seconds
- \( n = 7 \), total wait = 190 seconds
- \( n = 8 \), total wait = 220 seconds
- \( n = 9 \), total wait = 250 seconds
- \( n = 10 \), total wait = 280 seconds

If the Execution program encounters a DB2 STOP command, Execution checks the object status to see whether the object actually stopped. If not, the Execution program waits for the specified interval and checks again.

If \( n \) number of intervals passes without the object stopping successfully, the Execution program terminates with a -STOP command. If such a stop occurs, you can restart Execution when the object finally stops.

The default value is 3. A value of 0 indicates that if the object does not stop, the worklist stops without waiting. The maximum value allowed for this keyword is 10 (which is 280 seconds).
Table 57  AEXIN keywords  (Part 7 of 7)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOPWTSECS (n)</td>
<td>specifies the number (n) of seconds to wait in the first interval of the STOPWAIT keyword for a DB2 STOP command to stop a database or table space. By default the length of the first interval is 10 seconds. See also the STOPWAIT AEXIN keyword in this table.</td>
</tr>
<tr>
<td>SYNCDELETE</td>
<td>instructs Execution to remove all sync entries when an Execution job completes with no errors</td>
</tr>
<tr>
<td>SYNLIST</td>
<td>prints a synonym list</td>
</tr>
<tr>
<td>TASKID (a,b)</td>
<td>specifies the task ID to use</td>
</tr>
<tr>
<td>UNLOADDOPT</td>
<td>provides the BMC products with the name of the options module for the BMC UNLOAD PLUS product. The BMC_UNLOAD_OPTS keyword in the AJXPOFIN input stream replaces this keyword.</td>
</tr>
<tr>
<td>WARNRC</td>
<td>specifies the return code to use for warnings</td>
</tr>
<tr>
<td>WORKID (a,b)</td>
<td>specifies the work ID to use. Execution fails if this work ID does not match the work ID that the -WKID command in the worklist specifies. The AJXIN and ALUIN input streams also use this keyword.</td>
</tr>
</tbody>
</table>
Table 58 lists the keywords in the POF and the corresponding SLIB variable (or ISPF variable). The product uses the keywords in your initial POF and your user POF. If a default value is provided for a keyword, the value follows an equal sign after the keyword.

### Table 58  POF keywords (Part 1 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>2MEGSQNL=N</td>
<td>indicates whether to allocate a 2-MB buffer for large SQL statements</td>
<td>AJX2MGS</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>ACM_AMS=Y</td>
<td>for ALTER and CHANGE MANAGER, controls whether Analysis, by default, generates AMS statements (IDCAMS DELETE and DEFINE) in the worklist</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The following values are valid:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Y generates AMS statements (IDCAMS DELETE and DEFINE) in a worklist.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- N generates a -STOP worklist command that enables you to complete the DELETE and DEFINE commands before the DB2 object CREATE commands that are located later in the worklist.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>You can use the INCLUDE (AMS) keyword to override this value.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>ACM_BASDIAG=SYSOUT</td>
<td>for CHANGE MANAGER, sets the default value for the Baseline diagnostic output data set name</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>ACM_BRPTDIAG=SYSOUT</td>
<td>for CHANGE MANAGER, specifies the default name for the Baseline Report diagnostic output data set</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>ACM_BRPTDSN=’&amp;PREFIX..BASELINE.REPORT’</td>
<td>for CHANGE MANAGER, specifies the default name for the Baseline Report data set name</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>Keyword and default value</td>
<td>Description</td>
<td>SLIB variable</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>ACM_CDLDSN=&amp;PREFIX..&amp;SSID..CDL(CDL)’</td>
<td>for CHANGE MANAGER, defines the default data set name for generated Change Definition Language (CDL) statements. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_CDLPS=15</td>
<td>for CHANGE MANAGER, defines, in tracks, the default value for the primary space allocation of the CDL data set. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_CDLSS=5</td>
<td>for CHANGE MANAGER, defines, in tracks, the default value for the secondary space allocation of the CDL data set. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_CDLU=SYSDA</td>
<td>for CHANGE MANAGER, defines the default unit for the CDL data set. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_CMPDIAG=SYSOUT</td>
<td>for CHANGE MANAGER, defines the default value for the Compare diagnostic output data set. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_CPLCDLO=&amp;PREFIX..&amp;SSID..CDL(CDL)’</td>
<td>for the CM/PILOT component of CHANGE MANAGER, this keyword defines the default data set name for generated CDL statements. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_CPLDIAG=SYSOUT</td>
<td>for CHANGE MANAGER, specifies the default name for the CM/PILOT component’s diagnostic output data set. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 3 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
</table>
| ACM_CPLWDSN=’&PREFIX..&SSID..&TASKID’ | for CHANGE MANAGER, specifies the default worklist data set name for a new TASKID used in the CM/PILOT component  

CHANGE MANAGER dynamically allocates the data set the first time that the data set is used. This data set can be either a sequential file or a partitioned data set (PDS).  

This keyword is not included in the AJXPOFIN input stream. | none |
| ACM_CPLWDSNO=’&PREFIX..&SSID..&WORKID’ | for the CM/PILOT component of CHANGE MANAGER, this keyword specifies the default worklist data set name for a work ID  

CHANGE MANAGER dynamically allocates the data set the first time that the data set is used. This data set can be either a sequential file or a partitioned data set (PDS).  

This keyword is not included in the AJXPOFIN input stream. | none |
| ACM_DBRM1                        | for ALTER and CHANGE MANAGER, specifies the name of a default DBRM library  

This keyword is not included in the AJXPOFIN input stream. | none |
| ACM_DBRM2                        | for ALTER and CHANGE MANAGER, specifies the name of a default DBRM library  

This keyword is not included in the AJXPOFIN input stream. | none |
| ACM_DBRM3                        | for ALTER and CHANGE MANAGER, specifies the name of a default DBRM library  

This keyword is not included in the AJXPOFIN input stream. | none |
| ACM_GLID=id                      | for ALTER and CHANGE MANAGER, defines a global authorization ID (GLID)  

This authorization ID is used instead of the authorization ID of the person who submits the Execution job. The worklist begins with a -GLID command that switches authorization to the GLID.  

This keyword is not included in the AJXPOFIN input stream. | none |
<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM_IMPDIAG=SYSOUT</td>
<td>for ALTER and CHANGE MANAGER, defines the default name for the Import diagnostic output data set. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_JDSN=</td>
<td>for ALTER and CHANGE MANAGER, defines the default data set name that is used for Analysis JCL. This data set can be either a sequential or a partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. The products automatically use the work ID as the member name. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_JDSNB=</td>
<td>for CHANGE MANAGER, defines the default data set name that is used for Baseline JCL. This data set can be either a sequential or a partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. CHANGE MANAGER automatically uses the work ID as the member name. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_JDSNBG=</td>
<td>for ALTER and CHANGE MANAGER, defines the default data set name that is used for Batch Execution JCL Generation. This data set can be either a sequential or a partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. The products automatically use the work ID as the member name. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_JDSNBR=</td>
<td>for CHANGE MANAGER, specifies the default data set name where the product places the generated Baseline Report JCL. This data set can be either a sequential or partitioned data set. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
</tbody>
</table>
### Table 58   POF keywords (Part 5 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM_JDSNC= ' &amp;PREFIX..COMPARE(CMPJCL)'</td>
<td>for CHANGE MANAGER, defines the default data set name that is used for Compare JCL. This data set can be either a sequential or partitioned data set. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_JDSNCPL= ' &amp;PREFIX..TASKID(&amp;TASKID)'</td>
<td>for CHANGE MANAGER, specifies the default data set name where the CM/PILOT component places the generated Execution JCL. This data set can be either a sequential or partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. CM/PILOT automatically uses the task ID as the member name. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_JDSNCPLO= ' &amp;PREFIX..EXEC(&amp;WORKID)'</td>
<td>for the CM/PILOT component of CHANGE MANAGER, this keyword defines the default data set name that is used for Execution JCL. This data set can be either a sequential or partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. The product automatically uses the work ID as the member name. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_JDSNE= ' &amp;PREFIX..EXEC(&amp;WORKID)'</td>
<td>for ALTER and CHANGE MANAGER, defines the default data set name that is used for Execution JCL. This data set can be either a sequential or partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. The products automatically use the work ID as the member name. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
<tr>
<td>ACM_JDSNI= ' &amp;PREFIX..IMPORT(&amp;WORKID)'</td>
<td>for ALTER and CHANGE MANAGER, defines the default data set name that is used for Import JCL. This data set can be either a sequential or partitioned data set. Hardcoding a member name is not recommended for a partitioned data set. The products automatically use the work ID as the member name. This keyword is not included in the AJXPOFIN input stream.</td>
<td>none</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 6 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM_PARALLEL_MAXINIT=3</td>
<td>for the Database Administration solution, specifies the maximum number of BMC Cross-System Image Manager (XIM) initiators to use when executing a worklist in parallel. This value controls the number of permanent work data sets that are allocated in the Execution JCL. The valid range of values is 1 through 32. The maximum number of initiators should not exceed the number of objects in a worklist.</td>
<td>AJXPMXI</td>
</tr>
<tr>
<td>ACM_PARALLEL_MININIT=2</td>
<td>for the Database Administration solution, specifies the minimum number of the XIM technology initiators to use when executing a worklist in parallel. If the minimum number of XIM initiators is not available, the worklist does not execute. The valid range of values is 1 through 8.</td>
<td>AJXPMNI</td>
</tr>
<tr>
<td>ACM_PARALLEL_WORKLST=N</td>
<td>for the Database Administration solution, indicates whether a CHANGE MANAGER worklist should be executed in parallel. Y executes the worklist in parallel. However, if adequate XIM technology resources are not available, the Execution function fails. In addition, if the required parallelism worklist commands (such as -BEGG and -ENDG) are not included in the worklist, the worklist is not executed in parallel. N executes the worklist sequentially, even if the required parallelism worklist commands are included in the worklist.</td>
<td>AJXPWRK</td>
</tr>
<tr>
<td>ACM_PARALLEL_XIMGRP=XIMACM</td>
<td>for the Database Administration solution, specifies the group name for the XIM technology. The group name for XIM must be unique for each instance of XIM that is running on an OS/390 or z/OS image.</td>
<td>AJXXGRP</td>
</tr>
<tr>
<td>ACM_PARALLEL_XIMPROC=XIMACM</td>
<td>for the Database Administration solution, specifies the name of the procedure that the solution uses to start the XIM technology automatically. BMC recommends that the name of the XIM started task procedure be unique for each instance of XIM that is running on an OS/390 or z/OS image.</td>
<td>AJXXPRC</td>
</tr>
<tr>
<td>ACM_PARALLEL_XIMSTRT=N</td>
<td>for the Database Administration solution, indicates whether the XIM technology should be started automatically.</td>
<td>AJXXSTR</td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 7 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
</table>
| ACM_PARALLEL_XIMTRCE=N    | for the Database Administration solution, indicates whether tracing is used during the execution of a worklist  
  - Y writes tracing records to the AEXPTRAC output data set. AEXPTRAC is dynamically allocated, and the output is written to SYSOUT.  
  - N does not use tracing, even if an //AEXPTRAC DD statement is specified in the JCL. | AJXXTRC |
| ACM_PIC=N                 | for ALTER and CHANGE MANAGER, indicates whether an image copy should be taken of each table space before a database is dropped, a table is dropped, or the table space is dropped or reorganized.  
  This keyword is not included in the AJXPOFIN input stream. | none |
| ACM_SDSN=SYSOUT           | for ALTER and CHANGE MANAGER, specifies the default data set for diagnostic messages for Analysis  
  The value can be a sequential file, the keyword SYSOUT, or TERM (terminal).  
  - If you use SYSOUT, the diagnostic messages are written to the JES SPOOL.  
  - If you use TERM, the diagnostic messages are written to your terminal.  
  This keyword is not included in the AJXPOFIN input stream. | none |
| ACM_SDSNE=SYSOUT          | for ALTER and CHANGE MANAGER, specifies the default data set for diagnostic messages for Execution  
  The value can be a sequential file or the keyword SYSOUT. If you use SYSOUT, the diagnostic messages are written to the JES SPOOL.  
  This keyword is not included in the AJXPOFIN input stream. | none |
| ACM_WDSN=                  | for ALTER and CHANGE MANAGER, defines the default data set name for a worklist that Analysis generates  
  This keyword is not included in the AJXPOFIN input stream. | none |

'&PREFIX..&SSID..&WORKID'
Table 58  POF keywords (Part 8 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
</table>
| ACM_WLORDER              | for ALTER and CHANGE MANAGER, specifies how the Analysis component sorts objects in a worklist. The following values are valid:  
- C sorts the worklist by each table’s cardinality in descending sequence.  
- N sorts the worklist by table order in ascending sequence, according to the table owner and table name.  
- A sorts the worklist by either table cardinality or table order, depending on whether the worklist is processed in parallel.  
- Blank generates the objects in the worklist in an unsorted, random order.  
If the Database Administration solution processes the worklist in parallel, Analysis sorts the worklist by table cardinality. Otherwise, it sorts the worklist by table order.  
This keyword is not included in the AJXPOFIN input stream. | AJXOWLD |
| ACM_WLORDERMSG=Y         | for ALTER and CHANGE MANAGER, specifies whether to record in the SYSPRINT data set and in the worklist the amount of time to sort a worklist.  
This keyword is not included in the AJXPOFIN input stream. | AJXOWLM |
| ACM_WLPS=15              | for ALTER and CHANGE MANAGER, defines, in tracks, the default primary space allocation for the worklist.  
This keyword is not included in the AJXPOFIN input stream. | none |
| ACM_WLSS=5               | for ALTER and CHANGE MANAGER, defines, in tracks, the default secondary space allocation for the worklist.  
This keyword is not included in the AJXPOFIN input stream. | none |
| ACM_WLU=SYSDA            | for ALTER and CHANGE MANAGER, defines the default worklist unit.  
This keyword is not included in the AJXPOFIN input stream. | none |
| ADDLOAD1                 | defines an additional LINK library. | AJXJADD1 |
Table 58  POF keywords (Part 9 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDLOAD2=HLQ..UDBLINK</td>
<td>defines an override LINK library</td>
<td>AJXJADD2</td>
</tr>
<tr>
<td></td>
<td>This LINK library is placed first in any STEPLIB concatenation.</td>
<td></td>
</tr>
<tr>
<td>ARCH_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the archive data set</td>
<td>AJXARDCL</td>
</tr>
<tr>
<td>ARCH_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the archive data set (used if the threshold is exceeded)</td>
<td>AJXARTAD</td>
</tr>
<tr>
<td>ARCH_EXPDT</td>
<td>specifies the expiration date of the archive data set on tape</td>
<td>AJXAREX</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.</td>
<td></td>
</tr>
<tr>
<td>ARCH_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the archive data set</td>
<td>AJXARMCL</td>
</tr>
<tr>
<td>ARCH_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the archive data set (used if the threshold is exceeded)</td>
<td>AJXARTAM</td>
</tr>
<tr>
<td>ARCH_PREFIX=&amp;PREFIX..&amp;WKID</td>
<td>specifies the prefix for the name of the archive discard data set</td>
<td>AJXARPRF</td>
</tr>
<tr>
<td>ARCH_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the archive discard data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXACPS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>ARCH_RETPD</td>
<td>specifies the retention period for the archive data set on tape</td>
<td>AJXARRP</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td></td>
</tr>
<tr>
<td>ARCH_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the archive discard data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXACSS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>ARCH_STACK=N</td>
<td>specifies whether to stack the archive data set on a tape with data sets of the same type</td>
<td>AJXSTAKA</td>
</tr>
<tr>
<td>ARCH_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the archive data set</td>
<td>AJXARSCL</td>
</tr>
<tr>
<td>ARCH_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the archive data set (used if the threshold is exceeded)</td>
<td>AJXARTAS</td>
</tr>
<tr>
<td>Keyword and default value</td>
<td>Description</td>
<td>SLIB variable</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>ARCH_THRESH=0</td>
<td>specifies the maximum anticipated size for the archive data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td>AJXARTHX</td>
</tr>
<tr>
<td>ARCH_UNIT=SYSDA</td>
<td>specifies the unit for the archive discard data set</td>
<td>AJXACUNI</td>
</tr>
<tr>
<td>ARCH_UNIT_ALT</td>
<td>specifies the alternate unit name for the archive data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keyword.</td>
<td>AJXARALU</td>
</tr>
<tr>
<td>ASU_XP_LOGD_DATAC=</td>
<td>for DASD MANAGER PLUS, specifies the SMS data class and the allocation attributes of the Export log file</td>
<td>AJXXPDCL</td>
</tr>
<tr>
<td>ASU_XP_LOGD_MGMTC=</td>
<td>for DASD MANAGER PLUS, specifies the SMS management class that defines the migration, retention, and backup requirements of the Export log file</td>
<td>AJXXPMCL</td>
</tr>
<tr>
<td>ASU_XP_LOGD_PRIQTY=10</td>
<td>for DASD MANAGER PLUS, defines the primary allocation for the Export log file</td>
<td>AJXXPPS</td>
</tr>
<tr>
<td>ASU_XP_LOGD_SECQTY=2</td>
<td>for DASD MANAGER PLUS, defines the secondary allocation for the Export log file</td>
<td>AJXXPSS</td>
</tr>
<tr>
<td>ASU_XP_LOGD_STORC=</td>
<td>for DASD MANAGER PLUS, specifies the SMS storage class that defines the processing requirements of the Export log file</td>
<td>AJXXPSCL</td>
</tr>
<tr>
<td>ASU_XP_LOGD_UNIT= SYSDA</td>
<td>for DASD MANAGER PLUS, specifies the unit for the Export log file</td>
<td>AJXXPU</td>
</tr>
<tr>
<td>ASU_XP_LOGDSN=&amp;PREFIX.XPORT.LOG</td>
<td>for DASD MANAGER PLUS, specifies the Export log file</td>
<td>AJXXPLOG</td>
</tr>
<tr>
<td>ASU_XP_UIMSRVHOST=</td>
<td>for DASD MANAGER PLUS, specifies the host name of the primary UIM server for the Export utility</td>
<td>AJXXPSRV</td>
</tr>
<tr>
<td>ASU_XP_UIMSRVPORT=1</td>
<td>for DASD MANAGER PLUS, specifies the port number of the primary UIM server that contains the host definitions repository for the Export utility</td>
<td>AJXXPPT</td>
</tr>
<tr>
<td>ASU_XP_UIMSRVTIMEOUT=300</td>
<td>for DASD MANAGER PLUS, specifies the UIM timeout parameter that determines how long the Export utility should wait for a response from the UIM server before timing out</td>
<td>AJXXPTO</td>
</tr>
<tr>
<td>BINDFAIL=N</td>
<td>specifies whether worklist execution continues if a bind fails. If BINDFAIL=Y, worklist execution stops with a return code of 8. The stop is noted in the sync tables, and an Execution restart continues with the command that caused the failure. If BINDFAIL=N, worklist execution continues.</td>
<td>AJXBF</td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 11 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLRP_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the baseline recovery point data set</td>
<td>AJXBLDCL</td>
</tr>
<tr>
<td>BLRP_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the baseline recovery point data set (used if the threshold is exceeded)</td>
<td>AJXBLTAD</td>
</tr>
<tr>
<td>BLRP_EXPDT</td>
<td>specifies the expiration date of the baseline recovery point data set on tape</td>
<td>AJXBLEX</td>
</tr>
<tr>
<td></td>
<td>The valid values are yyddd or yyyy/ddd.</td>
<td></td>
</tr>
<tr>
<td>BLRP_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the baseline recovery point data set</td>
<td>AJXBLMCL</td>
</tr>
<tr>
<td>BLRP_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the baseline recovery point data set (used if the threshold is exceeded)</td>
<td>AJXBLTAM</td>
</tr>
<tr>
<td>BLRP_PREFIX=&amp;PREFIX.&amp;OBNOD</td>
<td>defines the high-level qualifier, or prefix, used for data sets containing data stored for a baseline recovery point</td>
<td>AJXBLPRF</td>
</tr>
<tr>
<td>BLRP_PRIQTY=10</td>
<td>defines the primary allocation quantity for baseline recovery point data sets if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXBRPS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 to 99999.</td>
<td></td>
</tr>
<tr>
<td>BLRP RETPD</td>
<td>specifies the retention period for the baseline recovery point data set on tape</td>
<td>AJXBLRP</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td></td>
</tr>
<tr>
<td>BLRP SECQTY=2</td>
<td>defines the secondary allocation quantity for baseline recovery point data sets if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXBRSS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>BLRP STACK=N</td>
<td>specifies whether to stack full-recovery baseline data sets on tape</td>
<td>AJXSTAKB</td>
</tr>
<tr>
<td>BLRP STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the baseline recovery point data set</td>
<td>AJXBLSCL</td>
</tr>
<tr>
<td>BLRP STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the baseline recovery point data set (used if the threshold is exceeded)</td>
<td>AJXBLTAS</td>
</tr>
<tr>
<td>BLRP THRESH=0</td>
<td>specifies the maximum anticipated size for the baseline recovery point data set, in cylinders</td>
<td>AJXBLTHX</td>
</tr>
<tr>
<td></td>
<td>If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td></td>
</tr>
<tr>
<td>Keyword and default value</td>
<td>Description</td>
<td>SLIB variable</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>BLRP_UNIT=SYSDA</td>
<td>specifies the unit used for the baseline recovery point data set</td>
<td>AJXBRUNI</td>
</tr>
<tr>
<td>BLRP_UNIT_ALT</td>
<td>specifies the alternate unit name for the baseline recovery point data set (used if the threshold value is exceeded) The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords.</td>
<td>AJXBLALU</td>
</tr>
<tr>
<td>BMC_CHECK_LOAD</td>
<td>specifies the name of the LINK library for the BMC CHECK PLUS utility This keyword replaces the CHECK+_LOAD AJXPOFIN keyword. If both BMC_CHECK_LOAD and CHECK+_LOAD are included in the POF, the components use the value that is specified for CHECK+_LOAD.</td>
<td>AJXJACKL</td>
</tr>
<tr>
<td>BMC_CHECK_OPTS=ACK$OPTS</td>
<td>specifies the name of the BMC CHECK PLUS utility installation options module This keyword replaces the CHECKDOPT AJXPOFIN keyword. If both BMC_CHECK_OPTS and CHECKDOPT are included in the POF, the components use the value that is specified for CHECKDOPT.</td>
<td>AJXKDOPT</td>
</tr>
<tr>
<td>BMC_COPY_LOAD</td>
<td>specifies the name of the LINK library for the BMC COPY PLUS utility This keyword replaces the COPY+_LOAD AJXPOFIN keyword. If both BMC_COPY_LOAD and COPY+_LOAD are included in the POF, the components use the value that is specified for COPY+_LOAD.</td>
<td>AJXJACPL</td>
</tr>
<tr>
<td>BMC_COPY_OPTS=ACP$OPTS</td>
<td>specifies the name of the BMC COPY PLUS utility installation options module This keyword replaces the COPYDOPT AJXPOFIN keyword. If both BMC_COPY_OPTS and COPYDOPT are included in the POF, the components use the value that is specified for COPYDOPT.</td>
<td>AJXPDOPT</td>
</tr>
<tr>
<td>BMC_LOAD_LOAD</td>
<td>specifies the name of the LINK library for the BMC LOADPLUS utility This keyword replaces the LOAD+_LOAD AJXPOFIN keyword. If both BMC_LOAD_LOAD and LOAD+_LOAD are included in the POF, the components use the value that is specified for LOAD+_LOAD.</td>
<td>AJXJAMUL</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 13 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BMC_LOAD_OPTS=AMU$OPTS</strong></td>
<td>specifies the name of the BMC LOADPLUS utility installation options module</td>
<td>AJXLDOPT</td>
</tr>
<tr>
<td></td>
<td>This keyword replaces the LOADDOPT AJXPOFIN keyword. If both BMC_LOAD_OPTS and LOADDOPT are included in the POF, the components use the value that is specified for LOADDOPT.</td>
<td></td>
</tr>
<tr>
<td><strong>BMC_RECOVER_LOAD</strong></td>
<td>specifies the name of the LINK library for the BMC RECOVER PLUS utility</td>
<td>AJXJAFRL</td>
</tr>
<tr>
<td></td>
<td>This keyword replaces the RECOVER+_LOAD AJXPOFIN keyword. If both BMC_RECOVER_LOAD and RECOVER+_LOAD are included in the POF, the components use the value that is specified for RECOVER+_LOAD.</td>
<td></td>
</tr>
<tr>
<td><strong>BMC_RECOVER_OPTS=AFR$OPTS</strong></td>
<td>specifies the name of the installation options module for the BMC RECOVER PLUS utility</td>
<td>AJXVDOPT</td>
</tr>
<tr>
<td></td>
<td>This keyword replaces the RECOVERDOPT AJXPOFIN keyword. If both BMC_RECOVER_OPTS and RECOVERDOPT are included in the POF, the components use the value that is specified for RECOVERDOPT.</td>
<td></td>
</tr>
<tr>
<td><strong>BMC_REORG_LOAD</strong></td>
<td>specifies the name of the LINK library for the BMC REORG PLUS utility</td>
<td>AJXJARUL</td>
</tr>
<tr>
<td></td>
<td>This keyword replaces the REORG+_LOAD AJXPOFIN keyword. If both BMC_REORG_LOAD and REORG+_LOAD are included in the POF, the components use the value that is specified for REORG+_LOAD.</td>
<td></td>
</tr>
<tr>
<td><strong>BMC_REORG_OPTS=ARU$OPTS</strong></td>
<td>specifies the name of the installation options module for the BMC REORG PLUS utility</td>
<td>AJXRDOPT</td>
</tr>
<tr>
<td></td>
<td>This keyword replaces the REORGDOPT AJXPOFIN keyword. If both BMC_REORG_OPTS and REORGDOPT are included in the POF, the components use the value that is specified for REORGDOPT.</td>
<td></td>
</tr>
</tbody>
</table>
## Table 58  POF keywords (Part 14 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BMC_REORG_XBMID</strong></td>
<td>specifies the BMC EXTENDED BUFFER MANAGER (XBM) subsystem (SSID) that the BMC REORG PLUS utility accesses when it uses XBM or the XBM SNAPSHOT UPGRADE FEATURE (SUF) to create a snapshot of the data sets to be reorganized. ALTER and CHANGE MANAGER use this value when they reorganize a table space by using an online reorg (SHRLEVEL CHANGE). CATALOG MANAGER and DASD MANAGER PLUS do not use this value. The value of the SSID can be from 1 to 8 characters long.</td>
<td>none</td>
</tr>
<tr>
<td><strong>BMC_UNLOAD_LOAD</strong></td>
<td>specifies the name of the LINK library for the BMC UNLOAD PLUS utility. This keyword replaces the UNLOAD+_LOAD AJXPOFIN keyword. If both BMC_UNLOAD_LOAD and UNLOAD+_LOAD are included in the POF, the components use the value that is specified for UNLOAD+_LOAD.</td>
<td>AJXJADUL</td>
</tr>
<tr>
<td><strong>BMC_UNLOAD_OPTS=ADUSOPTS</strong></td>
<td>specifies the name of the installation options module for the BMC UNLOAD PLUS utility. This keyword replaces the UNLOADDOPT AJXPOFIN keyword. If both BMC_UNLOAD_OPTS and UNLOADDOPT are included in the POF, the components use the value that is specified for UNLOADDOPT.</td>
<td>AJXNDOPT</td>
</tr>
<tr>
<td><strong>CAT_LOAD</strong></td>
<td>specifies the name of the LINK library for the BMC CATALOG MANAGER product</td>
<td>AJXJACTL</td>
</tr>
<tr>
<td><strong>CHECK+_LOAD</strong></td>
<td>specifies the name of the LINK library for the BMC CHECK PLUS utility. The BMC_CHECK_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_CHECK_LOAD and CHECK+_LOAD are included in the POF, the components use the value that is specified for CHECK+_LOAD.</td>
<td>none</td>
</tr>
<tr>
<td><strong>CHECKDOPT</strong></td>
<td>specifies the name of the installation options module for the BMC CHECK PLUS utility. The BMC_CHECK_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_CHECK_OPTS and CHECKDOPT are included in the POF, the components use the value that is specified for CHECKDOPT.</td>
<td>none</td>
</tr>
<tr>
<td><strong>CHGMAN_LOAD</strong></td>
<td>specifies the name of the LINK library for CHANGE MANAGER</td>
<td>AJXJACML</td>
</tr>
</tbody>
</table>
The cleanup job step, which deletes permanent work data sets, is only performed if the condition code that is returned from any previous job step is less than or equal to the code specified in CLEANUP_RC.

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEANUP_RC=4</td>
<td>specifies the value of the return code from the JCL cleanup job step</td>
<td>AJXCLNRC</td>
</tr>
<tr>
<td></td>
<td>The cleanup job step, which deletes permanent work data sets, is only performed if the condition code that is returned from any previous job step is less than or equal to the code specified in CLEANUP_RC.</td>
<td></td>
</tr>
<tr>
<td>CNTL_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the control data set</td>
<td>AJXCLDCL</td>
</tr>
<tr>
<td>CNTL_EXPDT</td>
<td>specifies the expiration date of the control data set on tape</td>
<td>AJXCLEX</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.</td>
<td></td>
</tr>
<tr>
<td>CNTL_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the control data set</td>
<td>AJXCLMCL</td>
</tr>
<tr>
<td>CNTL_PREFIX=&amp;PREFIX..&amp;WKID..&amp;SSID</td>
<td>specifies the prefix for the name of the control data set that the BMC UNLOAD PLUS utility uses</td>
<td>AJXCLPRF</td>
</tr>
<tr>
<td>CNTL_PRIQTY=1</td>
<td>specifies the primary allocation (in cylinders) for the control data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXCLPS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>CNTL_RETPD</td>
<td>specifies the retention period for the control data set on tape</td>
<td>AJXCLRD</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td></td>
</tr>
<tr>
<td>CNTL_SECQTY=1</td>
<td>specifies the secondary allocation (in cylinders) for the control data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXCLSS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>CNTL_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the control data set</td>
<td>AJXCLSCL</td>
</tr>
<tr>
<td>CNTL_UNIT=SYSDA</td>
<td>specifies the unit name for the control data set</td>
<td>AJXCLUNI</td>
</tr>
<tr>
<td>COPY+_LOAD</td>
<td>specifies the name of the LINK library for the BMC COPY PLUS utility</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The BMC_COPY_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_COPY_LOAD and COPY+_LOAD are included in the POF, the components use the value that is specified for COPY+_LOAD.</td>
<td></td>
</tr>
<tr>
<td>Keyword and default value</td>
<td>Description</td>
<td>SLIB variable</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>COPYDOPT</td>
<td>specifies the name of the installation options module for the BMC COPY PLUS utility</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The BMC_COPY_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_COPY_OPTS and COPYDOPT are included in the POF, the components use the value that is specified for COPYDOPT.</td>
<td></td>
</tr>
<tr>
<td>DASD_LOAD</td>
<td>specifies the name of the LINK library for the BMC DASD MANAGER PLUS product</td>
<td>AJXJASUL</td>
</tr>
<tr>
<td>DASD_DOPT</td>
<td>specifies the name of the installation options module for the BMC DASD MANAGER PLUS product</td>
<td>none</td>
</tr>
<tr>
<td>DATA_PACKER_LOAD</td>
<td>specifies the name of the LINK library for the BMC DATA PACKER® product</td>
<td>AJXDPKLL</td>
</tr>
<tr>
<td>DATASETSIZING=N</td>
<td>specifies the type of data set sizing</td>
<td>AJXESTI</td>
</tr>
<tr>
<td></td>
<td>The following values are valid:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- N does not perform data set sizing.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- C uses IBM RUNSTATS to perform data set sizing by using statistics from the DB2 catalog.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- B uses BMCSTATS to perform data set sizing by using the statistics from the BMC DASD MANAGER PLUS product tables.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- O physically and randomly samples the VSAM objects to estimate data set sizes.</td>
<td></td>
</tr>
<tr>
<td>DATAWK_NBR=4</td>
<td>for CATALOG MANAGER, specifies the number of DATAWK data sets that the IBM REORG utility conditionally uses for sorting data</td>
<td>AJXODWNM</td>
</tr>
<tr>
<td>DATAWK_UNIT=SYSDA</td>
<td>for CATALOG MANAGER, specifies the unit name of the DATAWK data set that the IBM REORG utility conditionally uses for sorting data</td>
<td>AJXDWUNI</td>
</tr>
<tr>
<td>DB2EXIT</td>
<td>specifies the name of the DB2 EXIT library</td>
<td>AJXSLD1</td>
</tr>
<tr>
<td>DB2LOAD</td>
<td>specifies the name of the DB2 LINK library</td>
<td>AJXSLD2</td>
</tr>
<tr>
<td>DEF_GDG_BASE=N</td>
<td>specifies whether to create the base of the GDG at JCL generation time</td>
<td>AJXGDGB</td>
</tr>
<tr>
<td>DEF_GDG_LIMIT=10</td>
<td>specifies the maximum number of GDG data sets that are allowed for primary copies</td>
<td>AJXDGDDL</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 255.</td>
<td></td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 17 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEF_GDG_NOSCR=N</td>
<td>specifies whether the base of a GDG is defined in the IDCAMS DEFINE command as EMPTY (NOSCR)</td>
<td>AJXNOSCR</td>
</tr>
<tr>
<td></td>
<td>If the GDG is defined as EMPTY (NOSCR), the operating system uncatalogs the generation data set when it reaches the maximum number of generation data sets to keep (LIMIT). Otherwise, if the GDG is defined as SCRATCH (SCR), the operating system scratches (deletes) the generation data set when the data set is uncataloged.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEF_GDG2_LIMIT=10</td>
<td>specifies the maximum number of GDG data sets that are allowed for recovery copies</td>
<td>AJXGDGL2</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 255.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIAG_MSGCLASS</td>
<td>identifies the SYSOUT class that the components use for reporting incorrect entries in the POF</td>
<td>AJXODMSG</td>
</tr>
<tr>
<td></td>
<td>The default value is blank, which indicates that a report is not generated when the product is invoked. The asterisk (*) is a valid value in batch mode.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISC_DATACLASS</td>
<td>specifies the SMS definition for the discard data set’s data class</td>
<td>AJXDCDCL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISC_DATACLASS_ALT</td>
<td>specifies the SMS definition for the discard data set’s data class (used if the threshold is exceeded)</td>
<td>AJXDCTAD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISC_EXPDT</td>
<td>specifies the expiration date of the discard data set on tape</td>
<td>AJXDCEX</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. The valid values are yyyy/ddd or yyyy/ddd.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISC_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the discard data set</td>
<td>AJXDCMCL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISC_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the discard data set (used if the threshold is exceeded)</td>
<td>AJXDCTAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISC_PREFIX=&amp;PREFIX..&amp;OBNOD</td>
<td>specifies the prefix for the name of the discard data set</td>
<td>AJXDCPRF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISC_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the discard data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXDCPS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 9999.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISC_RETPD</td>
<td>specifies the retention period for the discard data set on tape</td>
<td>AJXDCRP</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 18 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISC_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the discard data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXDCSS</td>
</tr>
<tr>
<td>DISC_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the discard data set</td>
<td>AJXDCSCL</td>
</tr>
<tr>
<td>DISC_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the discard data set (used if the threshold is exceeded)</td>
<td>AJXDCTAS</td>
</tr>
<tr>
<td>DISC_THRESH=0</td>
<td>specifies the maximum anticipated size for the discard data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td>AJXDCTHX</td>
</tr>
<tr>
<td>DISC_UNIT=SYSDA</td>
<td>specifies the unit for the discard data set</td>
<td>AJXDCUNI</td>
</tr>
<tr>
<td>DISC_UNIT_ALT</td>
<td>specifies the alternate unit name for the discard data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords.</td>
<td>AJXDCALU</td>
</tr>
<tr>
<td>DISP_ALLOW_POPUP=N</td>
<td>for ALTER and CHANGE MANAGER, specifies whether to display a dialog or a panel when the ZOOM (F4) key is pressed on an object name. Y indicates to display the object name in a dialog. If the name is too long to be displayed in a dialog, the product displays the name in a panel. N indicates to display the object name in a panel.</td>
<td>AJXSZAP</td>
</tr>
<tr>
<td>DISP_AUTO_TAB=+</td>
<td>for ALTER and CHANGE MANAGER, specifies whether to display an autotab character in front of an object name that is too long to be displayed. N indicates not to display an autotab character. Any character other than N displays as the autotab character.</td>
<td>AJXSZAT</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 19 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISP_LOCATION=M</td>
<td>for ALTER, CHANGE MANAGER, and DASD MANAGER PLUS, specifies the location of characters to be omitted in object names that are too long to be displayed</td>
<td>AJXSZTL</td>
</tr>
<tr>
<td></td>
<td>The following values are valid:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- B replaces characters at the left end (beginning) of the name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- M replaces characters in the middle of the name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- E replaces characters at the right end (end) of the name</td>
<td></td>
</tr>
<tr>
<td>DISP_OMIT_CHAR=&lt;&gt;</td>
<td>for ALTER, CHANGE MANAGER, and DASD MANAGER PLUS, specifies the characters that replace the beginning and end of a truncated string in an object name that is too long to be displayed</td>
<td>AJXSZTC</td>
</tr>
<tr>
<td>DISP_STATS=N</td>
<td>specifies whether to include comments that show the statistics that the components use to determine the sizes of the data sets in the generated JCL</td>
<td>AJXDBGSZ</td>
</tr>
<tr>
<td>DISP_VAR_DBUG=N</td>
<td>specifies whether to include the SLIB variables that JCL Generation uses to create the JCL, as well as their assigned values, in the generated JCL</td>
<td>AJXDBGVC</td>
</tr>
<tr>
<td>DROPR_NOIC=N</td>
<td>for ALTER, CATALOG MANAGER, and CHANGE MANAGER, specifies whether to invoke the Drop Recovery feature of the BMC CATALOG MANAGER product and drop an object:</td>
<td>AJXDRNIC</td>
</tr>
<tr>
<td></td>
<td>- N does not allow an object to be dropped if no image copies of the object exist.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Y allows an object to be dropped, even if no image copies of the object exist.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 20 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNCHECK44=N</td>
<td>specifies whether to verify that the prefix of a data set name contains 44 characters</td>
<td>AJXDSN44</td>
</tr>
<tr>
<td></td>
<td>Typically, JCL Generation verifies whether a prefix of a data set name contains 35 characters. Change the value of this keyword to Y for the following reasons:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- You are modifying an SLIB because the ddnames that BMC generated do not meet your environment’s standards.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- You are creating the name of a data set.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- The data set name does not refer to a generation data group (GDG).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If you set the value to Y, you must ensure that the data set names are unique.</td>
<td></td>
</tr>
<tr>
<td>ERR_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the error data set</td>
<td>AJXERDCL</td>
</tr>
<tr>
<td>ERR_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the error data set (used if the threshold is exceeded)</td>
<td>AJXERTAD</td>
</tr>
<tr>
<td>ERR_EXPDT</td>
<td>specifies the expiration date of the error data set on tape</td>
<td>AJXEREX</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.</td>
<td></td>
</tr>
<tr>
<td>ERR_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the error data set</td>
<td>AJXERMCL</td>
</tr>
<tr>
<td>ERR_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the error data set (used if the threshold is exceeded)</td>
<td>AJXERTAM</td>
</tr>
<tr>
<td>ERR_PREFIX=&amp;PREFIX..&amp;WKID..&amp;STEPN</td>
<td>specifies the prefix for the name of the error data set</td>
<td>AJXERPRF</td>
</tr>
<tr>
<td>ERR_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the error data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXERPS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>ERR_RETDP</td>
<td>specifies the retention period for the error data set on tape</td>
<td>AJXERRP</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 21 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERR_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the error data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXERSSS</td>
</tr>
<tr>
<td>ERR_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the error data set.</td>
<td>AJXERSCL</td>
</tr>
<tr>
<td>ERR_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the error data set (used if the threshold is exceeded).</td>
<td>AJXERSCL</td>
</tr>
<tr>
<td>ERR_THRESH=0</td>
<td>specifies the maximum anticipated size for the error data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td>AJXERTHX</td>
</tr>
<tr>
<td>ERR_UNIT=SYSDA</td>
<td>specifies the unit for the error data set</td>
<td>AJXERERUNI</td>
</tr>
<tr>
<td>ERR_UNIT_ALT</td>
<td>specifies the alternate unit name for the error data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords.</td>
<td>AJXERALU</td>
</tr>
<tr>
<td>EXEC_LOAD</td>
<td>specifies the name of the Execution LINK library</td>
<td>AJXJAEXL</td>
</tr>
<tr>
<td>FCPY_DATACLASS</td>
<td>for CATALOG MANAGER, specifies the SMS definition for the data class associated with the flashcopy data set. The IBM® FlashCopy® feature creates the flashcopy (a point-in-time copy of a volume).</td>
<td>AJXFDC</td>
</tr>
<tr>
<td>FCPY_EXPDT</td>
<td>for CATALOG MANAGER, specifies the expiration date of the flashcopy data set on tape. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume). A data set cannot have an expiration date and a retention period. The valid values are yyyddd or yyyy/ddd.</td>
<td>AJXFEX</td>
</tr>
<tr>
<td>FCPY_MGMTCLASS</td>
<td>for CATALOG MANAGER, specifies the SMS definition for the management class associated with the flashcopy data set. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume).</td>
<td>AJXFMC</td>
</tr>
<tr>
<td>FCPY_PREFIX=&amp;PREFIX..&amp;OBNOD..P&amp;PART</td>
<td>for CATALOG MANAGER, specifies the prefix for the name of the flashcopy data set. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume).</td>
<td>AJXFPR</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 22 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCPY_PRIQTY=10</td>
<td>for CATALOG MANAGER, specifies the primary allocation (in cylinders) for the flashcopy data set if DATASETSIZING=N or if an error in sizing occurs. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume). Valid values are 1 through 99999.</td>
<td>AJXFPS</td>
</tr>
<tr>
<td>FCPY_RETPD</td>
<td>for CATALOG MANAGER, specifies the retention period for the flashcopy data set on tape. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume).</td>
<td>AJXFRP</td>
</tr>
<tr>
<td>FCPY_SECQTY=2</td>
<td>for CATALOG MANAGER, specifies the secondary allocation (in cylinders) for the flashcopy data set if DATASETSIZING=N or if an error in sizing occurs. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume). Valid values are 1 through 99999.</td>
<td>AJXFSS</td>
</tr>
<tr>
<td>FCPY_STORCLASS</td>
<td>for CATALOG MANAGER, specifies the SMS definition for the storage class associated with the flashcopy data set. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume).</td>
<td>AJXFSC</td>
</tr>
<tr>
<td>FCPY_SUPPRESS_SUFF=N</td>
<td>for CATALOG MANAGER, specifies whether to suppress adding the DD name to the end of the prefix for the flashcopy data set (Y or N). If you specify Y, you must ensure that the data set name is unique. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume).</td>
<td>AJXFSF</td>
</tr>
<tr>
<td>FCPY_UNIT=SYSDA</td>
<td>for CATALOG MANAGER, specifies the unit name for the flashcopy data set. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume).</td>
<td>AJXFPU</td>
</tr>
<tr>
<td>FILT_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the filter data set.</td>
<td>AJXFLDCL</td>
</tr>
<tr>
<td>FILT_EXPDT</td>
<td>specifies the expiration date of the filter data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyyy/dd or yyyy/dd.</td>
<td>AJXFLEX</td>
</tr>
<tr>
<td>FILT_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the filter data set.</td>
<td>AJXFLMCL</td>
</tr>
<tr>
<td>FILT_PREFIX=&amp;PREFIX..&amp;WKID..&amp;STEPN</td>
<td>specifies the prefix for the name of the filter data set.</td>
<td>AJXFLPRF</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 23 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILT_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the filter data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXFLPS</td>
</tr>
<tr>
<td>FILT_RETPD</td>
<td>specifies the retention period for the filter data set on tape</td>
<td>AJXFLRP</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td></td>
</tr>
<tr>
<td>FILT_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the filter data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXFLSS</td>
</tr>
<tr>
<td>FILT_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the filter data set</td>
<td>AJXFLSCL</td>
</tr>
<tr>
<td>FILT_UNIT=SYSDA</td>
<td>specifies the unit name for the filter data set</td>
<td>AJXFLUNI</td>
</tr>
<tr>
<td>GDG_MODEL=SYS1.MODEL</td>
<td>specifies the name of the GDG model data set</td>
<td>AJXGDGMD</td>
</tr>
<tr>
<td>HASHFAIL=N</td>
<td>specifies whether Execution terminates the job if a hash failure, such as a changed or added statement, occurs in a worklist</td>
<td>AJXHF</td>
</tr>
<tr>
<td>HASHWARNRC=n</td>
<td>defines the return code (n) that the product sends back when the product finds only hash warnings</td>
<td>AJXOHWRC</td>
</tr>
<tr>
<td></td>
<td>Do not use 8 for this value.</td>
<td></td>
</tr>
<tr>
<td>IOALOAD1</td>
<td>specifies the name of a LINK library for the utility automation component of the BMC Database Performance for DB2 solution</td>
<td>none</td>
</tr>
<tr>
<td>IOALOAD2</td>
<td>specifies the name of a LINK library for the utility automation component of the BMC Database Performance for DB2 solution</td>
<td>none</td>
</tr>
<tr>
<td>JCLCLEANUP=N</td>
<td>specifies whether to generate a job step that automatically deletes many of the permanent work data sets that Execution creates</td>
<td>AJXOCLN</td>
</tr>
<tr>
<td>JCLLIB</td>
<td>specifies the name of a partitioned data set (PDS) that contains JCL to be included in a job, or the name of the PDS that specifies the cataloged procedures (PROCs) that are used for non-worklist JCL</td>
<td>AJXJBLIB</td>
</tr>
<tr>
<td>JES3=N</td>
<td>specifies whether JCL is to be generated for a JES3 system</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>JCL Generation no longer uses this keyword.</td>
<td></td>
</tr>
<tr>
<td>JOB_INCLUDE_MEMBER</td>
<td>specifies the name of a JCL member to be included at the end of a job</td>
<td>AJXSINC</td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 24 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB CARD1= //JOBC JOB (&amp;ZACCTNUM),&amp;PGMR',</td>
<td>defines the default job card statement that the components use when JCL Generation generates Analysis and Execution JCL</td>
<td>AJXJB1, AJXJB2, AJXJB3, AJXJB4, AJXJB5</td>
</tr>
<tr>
<td>JOB CARD2= // CLASS=A, MSGLEVEL=(1,1)</td>
<td>For DASD MANAGER PLUS, these keywords specify the BMCTRIG and batch report job cards.</td>
<td></td>
</tr>
<tr>
<td>JOB CARD3= /*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOB CARD4= /*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOB CARD5= /*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LISTDEF DSN</td>
<td>for CATALOG MANAGER, specifies the name of the data set that contains member names for LISTDEF utility control statements</td>
<td>AJXOLDSN</td>
</tr>
<tr>
<td>LL_CLIB=CLIB</td>
<td>specifies the low-level qualifier (LLQ) for the product CLIST data sets for the batch ISPF environment</td>
<td>AJXLLC</td>
</tr>
<tr>
<td>LL_CLIB2</td>
<td>specifies the LLQ for the product CLIST data sets for the batch ISPF environment</td>
<td>AJXLLC2</td>
</tr>
<tr>
<td>LL_CLIB3</td>
<td>specifies the LLQ for the product CLIST data sets for the batch ISPF environment</td>
<td>AJXLLC3</td>
</tr>
<tr>
<td>LL_CLIB4</td>
<td>specifies the LLQ for the product CLIST data sets for the batch ISPF environment</td>
<td>AJXLLC4</td>
</tr>
<tr>
<td>LL_CLIB5</td>
<td>specifies the LLQ for the product CLIST data sets for the batch ISPF environment</td>
<td>AJXLLC5</td>
</tr>
<tr>
<td>LL_LINK=LINK</td>
<td>specifies the LLQ for the LOAD library data sets for the batch ISPF environment</td>
<td>AJXLLL</td>
</tr>
<tr>
<td>LL_LINK2</td>
<td>specifies the LLQ for the LOAD library data sets for the batch ISPF environment</td>
<td>AJXLLL2</td>
</tr>
<tr>
<td>LL_LINK3</td>
<td>specifies the LLQ for the LOAD library data sets for the batch ISPF environment</td>
<td>AJXLLL3</td>
</tr>
<tr>
<td>LL_LINK4</td>
<td>specifies the LLQ for the LOAD library data sets for the batch ISPF environment</td>
<td>AJXLLL4</td>
</tr>
<tr>
<td>LL_LINK5</td>
<td>specifies the LLQ for the LOAD library data sets for the batch ISPF environment</td>
<td>AJXLLL5</td>
</tr>
<tr>
<td>LL_MLIB=MLIB</td>
<td>specifies the LLQ for the message data sets for the batch ISPF environment</td>
<td>AJXLLM</td>
</tr>
<tr>
<td>LL_MLIB2</td>
<td>specifies the LLQ for the message data sets for the batch ISPF environment</td>
<td>AJXLLM2</td>
</tr>
<tr>
<td>LL_MLIB3</td>
<td>specifies the LLQ for the message data sets for the batch ISPF environment</td>
<td>AJXLLM3</td>
</tr>
<tr>
<td>LL_MLIB4</td>
<td>specifies the LLQ for the message data sets for the batch ISPF environment</td>
<td>AJXLLM4</td>
</tr>
<tr>
<td>LL_MLIB5</td>
<td>specifies the LLQ for the message data sets for the batch ISPF environment</td>
<td>AJXLLM5</td>
</tr>
<tr>
<td>LL_PLIB=PLIB</td>
<td>specifies the LLQ for the panel and Help library data sets for the batch ISPF environment</td>
<td>AJXLLP</td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 25 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>LL_PLIB2</td>
<td>specifies the LLQ for the panel and Help library data sets for the batch ISPF environment</td>
<td>AJXLLP2</td>
</tr>
<tr>
<td>LL_PLIB3</td>
<td>specifies the LLQ for the panel and Help library data sets for the batch ISPF environment</td>
<td>AJXLLP3</td>
</tr>
<tr>
<td>LL_PLIB4</td>
<td>specifies the LLQ for the panel and Help library data sets for the batch ISPF environment</td>
<td>AJXLLP4</td>
</tr>
<tr>
<td>LL_PLIB5</td>
<td>specifies the LLQ for the panel and Help library data sets for the batch ISPF environment</td>
<td>AJXLLP5</td>
</tr>
<tr>
<td>LL_SLIB=SLIB</td>
<td>specifies the LLQ for the ISPF skeleton data sets for the batch ISPF environment</td>
<td>AJXLLS</td>
</tr>
<tr>
<td>LL_SLIB2</td>
<td>specifies the LLQ for the ISPF skeleton data sets for the batch ISPF environment</td>
<td>AJXLLS2</td>
</tr>
<tr>
<td>LL_SLIB3</td>
<td>specifies the LLQ for the ISPF skeleton data sets for the batch ISPF environment</td>
<td>AJXLLS3</td>
</tr>
<tr>
<td>LL_SLIB4</td>
<td>specifies the LLQ for the ISPF skeleton data sets for the batch ISPF environment</td>
<td>AJXLLS4</td>
</tr>
<tr>
<td>LL_SLIB5</td>
<td>specifies the LLQ for the ISPF skeleton data sets for the batch ISPF environment</td>
<td>AJXLLS5</td>
</tr>
<tr>
<td>LL_TLIB=TLIB</td>
<td>specifies the LLQ for the ISPF table data sets for the batch ISPF environment</td>
<td>AJXLLT</td>
</tr>
<tr>
<td>LL_TLIB2</td>
<td>specifies the LLQ for the ISPF table data sets for the batch ISPF environment</td>
<td>AJXLLT2</td>
</tr>
<tr>
<td>LL_TLIB3</td>
<td>specifies the LLQ for the ISPF table data sets for the batch ISPF environment</td>
<td>AJXLLT3</td>
</tr>
<tr>
<td>LL_TLIB4</td>
<td>specifies the LLQ for the ISPF table data sets for the batch ISPF environment</td>
<td>AJXLLT4</td>
</tr>
<tr>
<td>LL_TLIB5</td>
<td>specifies the LLQ for the ISPF table data sets for the batch ISPF environment</td>
<td>AJXLLT5</td>
</tr>
<tr>
<td>LL_XML=XML</td>
<td>specifies the LLQ for the utility generation data sets for the batch ISPF environment</td>
<td>AJXLLX</td>
</tr>
<tr>
<td>LL_XML2</td>
<td>specifies the LLQ for the utility generation data sets for the batch ISPF environment</td>
<td>AJXLLX2</td>
</tr>
<tr>
<td>LL_XML3</td>
<td>specifies the LLQ for the utility generation data sets for the batch ISPF environment</td>
<td>AJXLLX3</td>
</tr>
<tr>
<td>LL_XML4</td>
<td>specifies the LLQ for the utility generation data sets for the batch ISPF environment</td>
<td>AJXLLX4</td>
</tr>
<tr>
<td>LL_XML5</td>
<td>specifies the LLQ for the utility generation data sets for the batch ISPF environment</td>
<td>AJXLLX5</td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 26 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLQ</td>
<td>specifies the LLQ for ISPF data sets for the batch ISPF environment</td>
<td>&amp;AJXLLQ</td>
</tr>
<tr>
<td></td>
<td>During installation, if you chose to use the runtime enablement feature, the Installation System set this value to BMC. If you chose not to use the feature, the Installation System set the value to DB.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Changing the qualifier of the ISPF data sets might cause unpredictable results. Do not change the LLQ for the ISPF data sets.</td>
<td></td>
</tr>
<tr>
<td>LOAD+_LOAD</td>
<td>specifies the name of the LINK library for the BMC LOADPLUS utility</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The BMC_LOAD_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_LOAD_LOAD and LOAD+_LOAD are included in the POF, the components use the value that is specified for LOAD+_LOAD.</td>
<td></td>
</tr>
<tr>
<td>LOADDOPT</td>
<td>specifies the name of the installation options module for the BMC LOADPLUS utility</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The BMC_LOAD_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_LOAD_OPTS and LOADDOPT are included in the POF, the components use the value that is specified for LOADDOPT.</td>
<td></td>
</tr>
<tr>
<td>LOGWK_NBR=4</td>
<td>specifies the number of LOGSORT data sets</td>
<td>AJXOLGNM</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 32.</td>
<td></td>
</tr>
<tr>
<td>LOGWK_UNIT=SYSDA</td>
<td>specifies the unit name of the LOGSORT data set</td>
<td>AJXLGSUT</td>
</tr>
<tr>
<td>MAP_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the map data set</td>
<td>AJXMPDCL</td>
</tr>
<tr>
<td>MAP_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the map data set (used if the threshold is exceeded)</td>
<td>AJXMPTAD</td>
</tr>
<tr>
<td>MAP_EXPDT</td>
<td>specifies the expiration date of the map data set on tape</td>
<td>AJXMPEX</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.</td>
<td></td>
</tr>
<tr>
<td>MAP_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the map data set</td>
<td>AJXPMCL</td>
</tr>
<tr>
<td>MAP_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the map data set (used if the threshold is exceeded)</td>
<td>AJXMPTAM</td>
</tr>
<tr>
<td>MAP_PREFIX=&amp;PREFIX..&amp;WKID..&amp;SSID</td>
<td>specifies the prefix for the name of the map data set</td>
<td>AJXMPPRF</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 27 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the map data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXMPPS</td>
</tr>
<tr>
<td>MAP_RETPD</td>
<td>specifies the retention period for the map data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td>AJXMPRP</td>
</tr>
<tr>
<td>MAP_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the map data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXMPSS</td>
</tr>
<tr>
<td>MAP_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the map data set</td>
<td>AJXMPSCl</td>
</tr>
<tr>
<td>MAP_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the map data set (used if the threshold is exceeded)</td>
<td>AJXMPtas</td>
</tr>
<tr>
<td>MAP_THRESH=0</td>
<td>specifies the maximum anticipated size for the map data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td>AJXMPThx</td>
</tr>
<tr>
<td>MAP_UNIT=SYSDA</td>
<td>specifies the unit for the map data set</td>
<td>AJXMPUNI</td>
</tr>
<tr>
<td>MAP_UNIT_ALT</td>
<td>specifies the alternate unit name for the map data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords.</td>
<td>AJXMPalu</td>
</tr>
<tr>
<td>MAX_CYL=99999</td>
<td>specifies the maximum number of cylinders in the data set. If this value is exceeded for a data set, MAX_PRIQTY and MAX_SECQTY are used for any type of data set that does not have a specific threshold limit specified. Valid values are 1 through 99999.</td>
<td>AJXMAXQT</td>
</tr>
<tr>
<td>MAX_PRIQTY=2000</td>
<td>specifies the primary quantity in cylinders that will be used when the value of MAX_CYL is reached. Valid values are 1 through 9999.</td>
<td>AJXMAXPR</td>
</tr>
</tbody>
</table>
## Table 58  POF keywords (Part 28 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX_SECQTY=200</td>
<td>specifies the secondary quantity in cylinders that will be used when the value of MAX_CYL is reached. Valid values are 1 through 9999.</td>
<td>AJXMAXSS</td>
</tr>
<tr>
<td>MAX_UNITCNT</td>
<td>specifies the value for the DASD unit count. Valid values are 1 through 59.</td>
<td>AJXUC</td>
</tr>
<tr>
<td>MEMLIMIT</td>
<td>specifies the limit on above-the-bar memory for an address space</td>
<td>AJXMLMT</td>
</tr>
<tr>
<td>ORTPARM_DSN</td>
<td>specifies the name of the data set for the SyncSort parameters</td>
<td>AJXSORTP</td>
</tr>
<tr>
<td>PCPY1_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the local primary copy data set</td>
<td>AJXC1DCL</td>
</tr>
<tr>
<td>PCPY1_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the local primary copy data set (used if the threshold is exceeded)</td>
<td>AJXCITAD</td>
</tr>
<tr>
<td>PCPY1_EXPDT</td>
<td>specifies the expiration date of the local primary copy data set on tape. A data set cannot have an expiration date and a retention period. The valid values are <strong>yyddd</strong> or <strong>yyyy/ddd</strong>.</td>
<td>AJXC1EX</td>
</tr>
<tr>
<td>PCPY1_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the local primary copy data set</td>
<td>AJXC1MCL</td>
</tr>
<tr>
<td>PCPY1_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the local primary copy data set (used if the threshold is exceeded)</td>
<td>AJXCITAM</td>
</tr>
<tr>
<td>PCPY1_PREFIX=&amp;PREFIX..&amp;OBNOD..P&amp;PART</td>
<td>specifies the prefix for the name of the local primary copy data set</td>
<td>AJXC1PRF</td>
</tr>
<tr>
<td>PCPY1_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the local primary copy data set if <strong>DATASETSIZING=N</strong> or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXSCPS</td>
</tr>
<tr>
<td>PCPY1_RETPD</td>
<td>specifies the retention period for the local primary copy data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 to 9999.</td>
<td>AJXC1RP</td>
</tr>
<tr>
<td>PCPY1_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the local primary copy data set if <strong>DATASETSIZING=N</strong> or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXSCSS</td>
</tr>
<tr>
<td>PCPY1_STACK=N</td>
<td>specifies whether to stack the local primary copy data set on a tape with data sets of the same type</td>
<td>AJXSTAKC</td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 29 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCPY1_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the local primary copy data set</td>
<td>AJXC1SCL</td>
</tr>
<tr>
<td>PCPY1_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the local primary copy data set (used if the threshold is exceeded)</td>
<td>AJXC1TAS</td>
</tr>
</tbody>
</table>
| PCPY1_SUPPRESS_SUFF=N                      | specifies whether to suppress adding the DD name to the end of the prefix for the local primary copy data set  
If you specify Y, you must ensure that the data set name is unique. | AJXC1SF       |
| PCPY1_THRESH=0                             | specifies the maximum anticipated size for the local primary copy data set, in cylinders  
If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used. | AJXC1THX      |
| PCPY1_UNIT=SYSDA                           | specifies the unit for the local primary copy data set                      | AJXSCUNI      |
| PCPY1_UNIT_ALT                             | specifies the alternate unit name for the local primary copy data set (used if the threshold value is exceeded)  
The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. | AJXC1ALU      |
| PCPY2_DATACLASS                            | specifies the SMS definition for the data class associated with the local backup copy data set | AJXC2DCL      |
| PCPY2_DATACLASS_ALT                        | specifies the SMS definition for the data class associated with the local backup copy data set (used if the threshold is exceeded) | AJXC2TAD      |
| PCPY2_EXPDT                                | specifies the expiration date of the local backup copy data set on tape  
A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd. | AJXC2EX       |
| PCPY2_MGMTCLASS                            | specifies the SMS definition for the management class associated with the local backup copy data set | AJXC2MCL      |
| PCPY2_MGMTCLASS_ALT                        | specifies the SMS definition for the management class associated with the local backup copy data set (used if the threshold is exceeded) | AJXC2TAM      |
| PCPY2_PREFIX= &PREFIX..&OBNOX..P&PART      | specifies the prefix for the name of the local backup copy data set          | AJXC2PRF      |
| PCPY2_PRIQTY=10                            | specifies the primary allocation (in cylinders) for the local backup copy data set if DATASETSIZING=N or if an error in sizing occurs  
Valid values are 1 through 99999. | AJXC2PS       |
Table 58  POF keywords (Part 30 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
</table>
| PCPY2_RETPD               | specifies the retention period for the local backup copy data set on tape  
A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999. | AJXC2RP |
| PCPY2_SECQTY=2            | specifies the secondary allocation (in cylinders) for the local backup copy data set if DATASETSIZING=N or if an error in sizing occurs  
Valid values are 1 through 9999. | AJXC2SS |
| PCPY2_STACK=N             | specifies whether to stack the local backup copy data set on a tape with data sets of the same type | AJXC2STK |
| PCPY2_STORCLASS           | specifies the SMS definition for the storage class associated with the local backup copy data set | AJXC2SCL |
| PCPY2_STORCLASS_ALT       | specifies the SMS definition for the storage class associated with the local backup copy data set (used if the threshold is exceeded) | AJXC2TAS |
| PCPY2_SUPPRESS_SUFF=N     | specifies whether to suppress adding the DD name to the end of the prefix for the local backup copy data set  
If you specify Y, you must ensure that the data set name is unique. | AJXC2SF |
| PCPY2_THRESH=0            | specifies the maximum anticipated size for the local backup copy data set, in cylinders  
If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used. | AJXC2THX |
| PCPY2_UNIT=SYSDA          | specifies the unit for the local backup copy data set | AJXC2UNI |
| PCPY2_UNIT_ALT            | specifies the alternate unit name for the local backup copy data set (used if the threshold value is exceeded)  
The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. | AJXC2ALU |
| POFDATE                   | shows the last date on which the initial POF was updated  
This value is created or updated when the POF is created or when it is updated by the AJXPOVAL or AJXPODAT edit macros. | none |
| PROC_BMCCHECK_NAME        | for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC CHECK PLUS for DB2 utility  
This keyword is not included in the AJXPOFIN input stream. | AJXPCHK |
### Table 58  POF keywords (Part 31 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC_BMCCHECK_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC CHECK PLUS for DB2 utility. This keyword is not included in the AJXPOFIN input stream.</td>
<td>AJXPKST</td>
</tr>
<tr>
<td>PROC_BMCCOPY_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC COPY PLUS for DB2 utility. This keyword is not included in the AJXPOFIN input stream.</td>
<td>AJXPCPY</td>
</tr>
<tr>
<td>PROC_BMCCOPY_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC COPY PLUS for DB2 utility. This keyword is not included in the AJXPOFIN input stream.</td>
<td>AJXPCST</td>
</tr>
<tr>
<td>PROC_BMCCPRS_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility. BMCCPRS references the utility that copies statistics from the DB2 catalog to the BMCSTATS tables. This keyword is not included in the AJXPOFIN input stream.</td>
<td>AJXPCPS</td>
</tr>
<tr>
<td>PROC_BMCCPRS_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility. BMCCPRS references the utility that copies statistics from the DB2 catalog to the BMCSTATS tables. This keyword is not included in the AJXPOFIN input stream.</td>
<td>AJXPZST</td>
</tr>
<tr>
<td>PROC_BMCLOAD_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC LOADPLUS for DB2 utility. This keyword is not included in the AJXPOFIN input stream.</td>
<td>AJXPLOD</td>
</tr>
<tr>
<td>Keyword and default value</td>
<td>Description</td>
<td>SLIB variable</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>PROC_BMCLOAD_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC LOADPLUS for DB2 utility</td>
<td>AJXPLST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCRECOVER_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC RECOVER PLUS for DB2 utility</td>
<td>AJXPRCV</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCRECOVER_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC RECOVER PLUS for DB2 utility</td>
<td>AJXPVST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCREORG_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC REORG PLUS for DB2 utility</td>
<td>AJXPREO</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCREORG_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC REORG PLUS for DB2 utility</td>
<td>AJXPRST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCSTATS_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC DASD MANAGER PLUS component of the BMC DASD MANAGER PLUS for DB2 utility</td>
<td>AJXPSTT</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCSTATS_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC DASD MANAGER PLUS component of the BMC DASD MANAGER PLUS for DB2 utility</td>
<td>AJXPSST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 33 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC_BMCSTOP_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility</td>
<td>AJXPSTP</td>
</tr>
<tr>
<td></td>
<td>BMCSTOP references the utility that issues a DB2 STOP command on an object and verifies the completion of the command.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCSTOP_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility</td>
<td>AJXPJST</td>
</tr>
<tr>
<td></td>
<td>BMCSTOP references the utility that issues a DB2 STOP command on an object and verifies the completion of the command.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCTRIG_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMCTRIG component of the BMC DASD MANAGER PLUS for DB2 utility</td>
<td>AJXPTRG</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCTRIG_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMCTRIG component of the BMC DASD MANAGER PLUS for DB2 utility</td>
<td>AJXPQST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCUNLOAD_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC UNLOAD PLUS for DB2 utility</td>
<td>AJXPULD</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCUNLOAD_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC UNLOAD PLUS for DB2 utility</td>
<td>AJXPUST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 34 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC_BMCUPRS_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility</td>
<td>AJXUPPRS</td>
</tr>
<tr>
<td></td>
<td>BMCUPRS references the utility that takes the statistics from the tables in the DASD MANAGER PLUS database (with the BMCSTATS utility) and updates the DB2 catalog with the statistics.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCUPRS_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility</td>
<td>AJXPYST</td>
</tr>
<tr>
<td></td>
<td>BMCUPRS references the utility that takes the statistics from the tables in the DASD MANAGER PLUS database (with the BMCSTATS utility) and updates the DB2 catalog with the statistics.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_DSNUTILB_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM DSNUTILB utility</td>
<td>AJXPDSN</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_DSNUTILB_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM DSNUTILB utility</td>
<td>AJXPDST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_DSN1COPY_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM DSN1COPY utility</td>
<td>AJXPDS1</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_DSN1COPY_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM DSN1COPY utility</td>
<td>AJXP1ST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 35 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC_GEN_SET_VAR=N</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies whether to include an SLIB that generates SET statements in the JCL for variables that you can use in catalog procedures (PROCs)</td>
<td>AJXPSETV</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_IDCAMSS_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM IDCAMS program</td>
<td>AJXPICM</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_IDCAMSS_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM IDCAMS program</td>
<td>AJXPIST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_IEBFR14_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM IEFBR14 job</td>
<td>AJXPB14</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_IEBFR14_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM IEFBR14 job</td>
<td>AJXPBST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_TSO_NAME</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for IBM TSO</td>
<td>AJXPTSO</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>PROC_TSO_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for IBM TSO</td>
<td>AJXPTST</td>
</tr>
<tr>
<td></td>
<td>This keyword is not included in the AJXPOFIN input stream.</td>
<td></td>
</tr>
<tr>
<td>Keyword and default value</td>
<td>Description</td>
<td>SLIB variable</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>PROC_USE=N</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies whether to generate a cataloged procedure (PROC) name instead of the EXEC PGM= statement for non-worklist JCL. This keyword is not included in the AJXPOFIN input stream.</td>
<td>AJXPUSE</td>
</tr>
<tr>
<td>PROC_USER_DEF_STEP</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for a user-defined program. This keyword is not included in the AJXPOFIN input stream.</td>
<td>AJXPFST</td>
</tr>
<tr>
<td>PROC_USER_DEFINED</td>
<td>for CATALOG MANAGER and DASD MANAGER PLUS, specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for a user-defined program. This keyword is not included in the AJXPOFIN input stream.</td>
<td>AJXPUDF</td>
</tr>
<tr>
<td>PUNCH_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the punch data set.</td>
<td>AJXPUDCL</td>
</tr>
<tr>
<td>PUNCH_EXPDT</td>
<td>specifies the expiration date of the punch data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.</td>
<td>AJXPUEX</td>
</tr>
<tr>
<td>PUNCH_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the punch data set.</td>
<td>AJXPUMCL</td>
</tr>
<tr>
<td>PUNCH_PREFIX= &amp;PREFIX..&amp;WKID..&amp;STEPN</td>
<td>when a limit key is changed, specifies the prefix of the data set that contains discard rows from the last partition of a table space after the table space is reorganized.</td>
<td>AJXPUPRF</td>
</tr>
<tr>
<td>PUNCH_PRIQTY=1</td>
<td>specifies the primary allocation (in cylinders) for the punch data set if DATASETSIZING=N or if a sizing error occurs. Valid values are 1 through 99999.</td>
<td>AJXPUPS</td>
</tr>
<tr>
<td>PUNCH_RETPD</td>
<td>specifies the retention period for the punch data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td>AJXPURP</td>
</tr>
</tbody>
</table>
**Table 58  POF keywords (Part 37 of 52)**

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUNCH_SECQTY=1</td>
<td>specifies the secondary allocation (in cylinders) for the punch data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXPUSS</td>
</tr>
<tr>
<td>PUNCH_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the punch data set</td>
<td>AJXPUSCL</td>
</tr>
<tr>
<td>PUNCH_UNIT=SYSDA</td>
<td>specifies the unit name for the punch data set</td>
<td>AJXPUUNI</td>
</tr>
<tr>
<td>RCPY1_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the recovery primary copy data set</td>
<td>AJXR1DCL</td>
</tr>
<tr>
<td>RCPY1_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the recovery primary copy data set (used if the threshold is exceeded)</td>
<td>AJXR1TAD</td>
</tr>
<tr>
<td>RCPY1_EXPDT</td>
<td>specifies the expiration date of the recovery primary copy data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.</td>
<td>AJXR1EX</td>
</tr>
<tr>
<td>RCPY1_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the recovery primary copy data set</td>
<td>AJXR1MCL</td>
</tr>
<tr>
<td>RCPY1_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the recovery primary copy data set (used if the threshold is exceeded)</td>
<td>AJXR1TAM</td>
</tr>
<tr>
<td>RCPY1_PREFIX= &amp;PREFIX..&amp;OBNOD..P&amp;PART</td>
<td>specifies the prefix for the name of the recovery primary copy data set</td>
<td>AJXR1PRF</td>
</tr>
<tr>
<td>RCPY1_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the recovery primary copy data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXR1CPS</td>
</tr>
<tr>
<td>RCPY1_RETPD</td>
<td>specifies the retention period for the recovery primary copy data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td>AJXR1RP</td>
</tr>
<tr>
<td>RCPY1_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the recovery primary copy data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXRCSS</td>
</tr>
<tr>
<td>RCPY1_STACK=N</td>
<td>specifies whether to stack the recovery primary copy data set on a tape with data sets of the same type</td>
<td>AJXSTAKR</td>
</tr>
<tr>
<td>RCPY1_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the recovery primary copy data set</td>
<td>AJXR1SCL</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 38 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCPY1_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the recovery primary copy data set (used if the threshold is exceeded)</td>
<td>AJXR1TAS</td>
</tr>
<tr>
<td>RCPY1_SUPPRESS_SUFF=N</td>
<td>specifies whether to suppress adding the DD name to the end of the prefix for the recovery primary copy data set. If you specify Y, you must ensure that the data set name is unique.</td>
<td>AJXR1SF</td>
</tr>
<tr>
<td>RCPY1_THRESH=0</td>
<td>specifies the maximum anticipated size for the recovery primary copy data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td>AJXR1THX</td>
</tr>
<tr>
<td>RCPY1_UNIT=SYSDA</td>
<td>specifies the unit for the recovery primary copy data set</td>
<td>AJXR1CUNI</td>
</tr>
<tr>
<td>RCPY1_UNIT_ALT</td>
<td>specifies the alternate unit name for the recovery primary copy data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords.</td>
<td>AJXR1ALU</td>
</tr>
<tr>
<td>RCPY2_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the recovery backup copy data set</td>
<td>AJXR2DCL</td>
</tr>
<tr>
<td>RCPY2_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the recovery backup copy data set (used if the threshold is exceeded)</td>
<td>AJXR2TAD</td>
</tr>
<tr>
<td>RCPY2_EXPDT</td>
<td>specifies the expiration date of the recovery backup copy data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyyyddd or yyyy/dddd.</td>
<td>AJXR2EX</td>
</tr>
<tr>
<td>RCPY2_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the recovery backup copy data set</td>
<td>AJXR2MCL</td>
</tr>
<tr>
<td>RCPY2_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the recovery backup copy data set (used if the threshold is exceeded)</td>
<td>AJXR2TAM</td>
</tr>
<tr>
<td>RCPY2_PREFIX=</td>
<td>specifies the prefix for the name of the recovery backup copy data set</td>
<td>AJXR2PRF</td>
</tr>
<tr>
<td>&amp;PREFIX..&amp;OBNOB..P&amp;PART</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCPY2_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the recovery backup copy data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXR2PS</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 39 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
</table>
| RCPY2_RETPD               | specifies the retention period for the recovery backup copy data set on tape  
A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999. | AJXR2RP |
| RCPY2_SECQTY=2            | specifies the secondary allocation (in cylinders) for the recovery backup copy data set if DATASETSIZING=N or if an error in sizing occurs  
Valid values are 1 through 99999. | AJXR2SS |
| RCPY2_STACK=N             | specifies whether to stack the recovery backup copy data set on a tape with data sets of the same type | AJXR2STK |
| RCPY2_STORCLASS           | specifies the SMS definition for the storage class associated with the recovery backup copy data set | AJXR2SCL |
| RCPY2_STORCLASS_ALT       | specifies the SMS definition for the storage class associated with the recovery backup copy data set (used if the threshold is exceeded) | AJXR2TAS |
| RCPY2_SUPPRESS_SUFF=N     | specifies whether to suppress adding the DD name to the end of the prefix for the recovery backup copy data set  
If you specify Y, you must ensure that the data set name is unique. | AJXR2SF |
| RCPY2_THRESH=0            | specifies the maximum anticipated size for the recovery backup copy data set, in cylinders  
If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used. | AJXR2THX |
| RCPY2_UNIT=SYSDA          | specifies the unit for the recovery backup copy data set | AJXR2UNI |
| RCPY2_UNIT_ALT            | specifies the alternate unit name for the recovery backup copy data set (used if the threshold value is exceeded)  
The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. | AJXR2ALU |
| REBINDFAIL=N              | specifies whether worklist execution continues if a rebind fails  
If REBINDFAIL=Y, worklist execution stops with a return code of 8. The stop is noted in the sync tables, and an Execution restart continues with the command that caused the failure. If REBINDFAIL=N, worklist execution continues. | AJXRBF |
<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>REBINDRC=n</strong></td>
<td>allows worklist execution to continue if a rebind fails, but returns the ( n ) value for a final condition code instead of 4, the default value</td>
<td>AJXRBC</td>
</tr>
<tr>
<td></td>
<td>Execution writes warning messages to AEXPRINT but does not post entries in the sync tables.</td>
<td></td>
</tr>
<tr>
<td><strong>RECOVER+_LOAD</strong></td>
<td>specifies the name of the LINK library for the BMC RECOVER PLUS utility</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The BMC_RECOVER_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_RECOVER_LOAD and RECOVER+_LOAD are included in the POF, the components use the value that is specified for RECOVER+_LOAD.</td>
<td></td>
</tr>
<tr>
<td><strong>RECOVERDOPT</strong></td>
<td>specifies the name of the installation options module for the BMC RECOVER PLUS utility</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The BMC_RECOVER_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_RECOVER_OPTS and RECOVERDOPT are included in the POF, the components use the value that is specified for RECOVERDOPT.</td>
<td></td>
</tr>
<tr>
<td><strong>REGION=0M</strong></td>
<td>defines the REGION parameter in the EXEC statement</td>
<td>AJXREGN</td>
</tr>
<tr>
<td><strong>REORG_MAPTAB</strong></td>
<td>specifies the name of the mapping table that the IBM REORG utility uses to map the row IDs (RIDs) in the source table to the RIDs in the target table</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>ALTER and CHANGE MANAGER use this value when reorganizing a table space by using an online reorganization (SHRLEVEL CHANGE). CATALOG MANAGER and DASD MANAGER PLUS do not use this value. The name can be from 1 to 72 characters long.</td>
<td></td>
</tr>
<tr>
<td><strong>REORG+_LOAD</strong></td>
<td>specifies the name of the LINK library for the BMC REORG PLUS utility</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The BMC_REORG_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_REORG_LOAD and REORG+_LOAD are included in the POF, the components use the value that is specified for REORG+_LOAD.</td>
<td></td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 41 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>REORGDOPT</td>
<td>specifies the name of the installation options module for the BMC REORG PLUS utility</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The BMC_REORG_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_REORG_OPTS and REORGDOPT are included in the POF, the components use the value that is specified for REORGDOPT.</td>
<td></td>
</tr>
<tr>
<td>REPT_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the report data set</td>
<td>AJXRPDCL</td>
</tr>
<tr>
<td>REPT_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the report data set (used if the threshold is exceeded)</td>
<td>AJXRPTAD</td>
</tr>
<tr>
<td>REPT_EXPDT</td>
<td>specifies the expiration date of the report data set on tape</td>
<td>AJXRPEX</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. The valid values are yyyyddd or yyyy/ddd.</td>
<td></td>
</tr>
<tr>
<td>REPT_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the report data set</td>
<td>AJXRPMCL</td>
</tr>
<tr>
<td>REPT_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the report data set (used if the threshold is exceeded)</td>
<td>AJXRPTAM</td>
</tr>
<tr>
<td>REPT_PREFIX=&amp;PREFIX..&amp;WKID</td>
<td>specifies the prefix for the name of the report data set</td>
<td>AJXRPPRF</td>
</tr>
<tr>
<td>REPT_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the report data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXRPPS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>REPT_RETPD</td>
<td>specifies the retention period for the report data set on tape</td>
<td>AJXRPRP</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td></td>
</tr>
<tr>
<td>REPT_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the report data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXRPPSS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>REPT_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the report data set</td>
<td>AJXRPSCL</td>
</tr>
<tr>
<td>REPT_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the report data set (used if the threshold is exceeded)</td>
<td>AJXRPTAS</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 42 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPT_THRESH=0</td>
<td>specifies the maximum anticipated size for the report data set, in cylinders</td>
<td>AJXRPTHX</td>
</tr>
<tr>
<td></td>
<td>If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td></td>
</tr>
<tr>
<td>REPT_UNIT=SYSDA</td>
<td>specifies the unit for the report data set</td>
<td>AJXRPUINI</td>
</tr>
<tr>
<td>REPT_UNIT_ALT</td>
<td>specifies the alternate unit name for the report data set, if the threshold value is exceeded</td>
<td>AJXRPAU</td>
</tr>
<tr>
<td></td>
<td>The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords.</td>
<td></td>
</tr>
<tr>
<td>RUNTIME_HLQ</td>
<td>specifies a high-level qualifier (HLQ) for ISPF data sets for the batch ISPF environment</td>
<td>&amp;AJXRHLQ</td>
</tr>
<tr>
<td></td>
<td>During installation, if you chose to use the runtime enablement feature, the Installation System set this value to an HLQ for user runtime libraries. If you chose not to use the feature, the Installation System set the value to an HLQ for Execution.</td>
<td></td>
</tr>
<tr>
<td>SORTWK_NBR=4</td>
<td>specifies the number of SORTWORK data sets</td>
<td>AJXSWNUM</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 32.</td>
<td></td>
</tr>
<tr>
<td>SORTWK_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the DATAWORK, LOGSORT, or SORTWORK data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXSWPS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>SORTWK_SECQTY=2</td>
<td>specifies the secondary quantity (in cylinders) for the DATAWORK, LOGSORT, or SORTWORK data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXSWSS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>SORTWK_UNIT=SYSDA</td>
<td>specifies the name of the SORTWORK data set</td>
<td>AJXSWUNI</td>
</tr>
<tr>
<td>SQLEXP_LOAD</td>
<td>specifies the name of the LINK library for the BMC SQL Explorer for DB2 product</td>
<td>AJXSQLL</td>
</tr>
<tr>
<td>SRTOUT_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the SRTOUT data set</td>
<td>AJXSODCCL</td>
</tr>
<tr>
<td>SRTOUT_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the SRTOUT data set (used if the threshold is exceeded)</td>
<td>AJXSOTAD</td>
</tr>
<tr>
<td>SRTOUT_EXPDT</td>
<td>specifies the expiration date of the SRTOUT data set on tape</td>
<td>AJXSOEX</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. The valid values are yyyy/dd or yyyy/gddd.</td>
<td></td>
</tr>
<tr>
<td>Keyword and default value</td>
<td>Description</td>
<td>SLIB variable</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>SRTOUT_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the SORTOUT data set</td>
<td>AJXSOMCL</td>
</tr>
<tr>
<td>SRTOUT_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the SORTOUT data set (used if the threshold is exceeded)</td>
<td>AJXSOTAM</td>
</tr>
<tr>
<td>SRTOUT_PREFIX=</td>
<td>specifies the prefix for the name of the SORTOUT data set</td>
<td>AJXSOPRF</td>
</tr>
<tr>
<td>&amp;PREFIX..&amp;WKID..&amp;STEPN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRTOUT_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the SORTOUT data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXSOPS</td>
</tr>
<tr>
<td>SRTOUT_RETPD</td>
<td>specifies the retention period for the SORTOUT data set on tape</td>
<td>AJXSORP</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td></td>
</tr>
<tr>
<td>SRTOUT_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the SORTOUT data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXSOSS</td>
</tr>
<tr>
<td>SRTOUT_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the SORTOUT data set</td>
<td>AJXSOSCL</td>
</tr>
<tr>
<td>SRTOUT_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the SORTOUT data set (used if the threshold is exceeded)</td>
<td>AJXSOTAS</td>
</tr>
<tr>
<td>SRTOUT_THRESH=0</td>
<td>specifies the maximum anticipated size for the SORTOUT data set, in cylinders</td>
<td>AJXSOTHX</td>
</tr>
<tr>
<td></td>
<td>If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td></td>
</tr>
<tr>
<td>SRTOUT_UNIT=SYSDA</td>
<td>specifies the unit for the SORTOUT data set</td>
<td>AJXSOUNI</td>
</tr>
<tr>
<td>SRTOUT_UNIT_ALT</td>
<td>specifies the alternate unit name for the SORTOUT data set, if the threshold value is exceeded</td>
<td>AJXSOALU</td>
</tr>
<tr>
<td></td>
<td>The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords.</td>
<td></td>
</tr>
<tr>
<td>STEP_INCLUDE_MEMBER</td>
<td>specifies the name of a JCL member to be included after each step in the JCL</td>
<td>AJXJINC</td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 44 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOPWAIT=3</td>
<td>specifies the number of intervals to wait for a DB2 STOP command to stop a database or table space. 0 indicates that if the object does not stop, the worklist stops without waiting. The maximum value allowed for this keyword is 10 (which is 280 seconds).</td>
<td>AJXSWT</td>
</tr>
<tr>
<td>STOPWTSECS=10</td>
<td>specifies the number of seconds to wait (during the first interval of the STOPWAIT keyword) for a DB2 STOP command to stop a database or table space.</td>
<td>AJXSWC</td>
</tr>
<tr>
<td>SUPPRESS_COMMENTS=N</td>
<td>specifies whether to suppress the comments in the generated JCL</td>
<td>AJXDBGSC</td>
</tr>
<tr>
<td>SYMDELETE=N</td>
<td>specifies whether Execution should remove all sync entries when an Execution job completes with no errors</td>
<td>AJXSYND</td>
</tr>
<tr>
<td>SYSEXEC</td>
<td>specifies the name of the partitioned data set in which a REXX EXEC is a member</td>
<td>AJXSYSEX</td>
</tr>
<tr>
<td>SYSTEM_MLIB</td>
<td>specifies the name of the system ISPF message library. You specify the value for this keyword during installation.</td>
<td>&amp;AJXSYSM</td>
</tr>
<tr>
<td>SYSUT_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the SYSUT data set</td>
<td>AJXSUDCL</td>
</tr>
<tr>
<td>SYSUT_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the SYSUT data set (used if the threshold is exceeded)</td>
<td>AJXSUTAD</td>
</tr>
<tr>
<td>SYSUT_EXPDT</td>
<td>specifies the expiration date of the SYSUT data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.</td>
<td>AJXSUEX</td>
</tr>
<tr>
<td>SYSUT_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the SYSUT data set</td>
<td>AJXSUMCL</td>
</tr>
<tr>
<td>SYSUT_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the SYSUT data set (used if the threshold is exceeded)</td>
<td>AJXSUTAM</td>
</tr>
<tr>
<td>SYSUT_PREFIX= &amp;PREFIX..&amp;WKID..&amp;STEPN</td>
<td>specifies the prefix for the name of the SYSUT and WORKDDN data sets</td>
<td>AJXSUPRF</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 45 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSUT_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the SYSUT data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXSUPS</td>
</tr>
<tr>
<td>SYSUT_RETPD</td>
<td>specifies the retention period for the SYSUT data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td>AJXSURP</td>
</tr>
<tr>
<td>SYSUT_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the SYSUT data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXSUSS</td>
</tr>
<tr>
<td>SYSUT_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the SYSUT data set</td>
<td>AJXSUSCL</td>
</tr>
<tr>
<td>SYSUT_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the SYSUT data set (used if the threshold is exceeded)</td>
<td>AJXSUTAS</td>
</tr>
<tr>
<td>SYSUT_THRESH=0</td>
<td>specifies the maximum anticipated size for the SYSUT data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td>AJXSUTHX</td>
</tr>
<tr>
<td>SYSUT_UNIT=SYSDA</td>
<td>specifies the unit for the SYSUT data set</td>
<td>AJXSUUNI</td>
</tr>
<tr>
<td>SYSUT_UNIT_ALT</td>
<td>specifies the alternate unit name for the SYSUT data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords.</td>
<td>AJXSUALU</td>
</tr>
<tr>
<td>SZDEVT=3390</td>
<td>specifies the device type used in data set sizing. Valid values are 3380 or 3390.</td>
<td>AJXODEV</td>
</tr>
<tr>
<td>TAPE_EXPDT</td>
<td>specifies the expiration date for the tape set. The value specified for this keyword is valid for all tape data sets for which an expiration date is not specified.</td>
<td>AJXEXPDT</td>
</tr>
<tr>
<td>TAPE_RETPD</td>
<td>specifies the retention period for the tape set. The value specified for this keyword is valid for all tape data sets for which a retention period is not specified.</td>
<td>AJXRETPD</td>
</tr>
</tbody>
</table>
Table 58  POF keywords (Part 46 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAPE_VOLCNT=99</td>
<td>specifies the maximum number of tape volumes Valid values are 0 through 255.</td>
<td>AJXVC</td>
</tr>
<tr>
<td>TAPE1</td>
<td>defines the names of the tape units for an installation</td>
<td>AJXTU1 AJXTU2 AJXTU3</td>
</tr>
<tr>
<td>TAPE2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TAPE3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEMPLATE_DSN</td>
<td>for CATALOG MANAGER, specifies the name of the data set that contains member names for TEMPLATE utility control statements</td>
<td>AJXOTDSN</td>
</tr>
<tr>
<td>TEMPUNIT=SYSDA</td>
<td>defines the name of the unit that the components use for temporary files</td>
<td>AJXWKUNI</td>
</tr>
<tr>
<td>TIMEPARM</td>
<td>indicates the time limit in minutes for each step in a batch job stream</td>
<td>AJXTIMEP</td>
</tr>
</tbody>
</table>
| TRTCH                     | specifies the parity, data conversion, translation, and compression for 7-track drives The following values are valid:  
  - Blank does not use a 7-track drive.  
  - C uses odd parity, conversion on, and translation off.  
  - E uses even parity, conversion off, and translation off.  
  - T uses odd parity, conversion off, and translation on.  
  - ET uses even parity, conversion off, and translation on.  
  - COMP uses data compression on.  
  - NOCOMP uses data compression off. | AJXTRTCH |
| TSOPROGRAM                | specifies an alternate TSO monitor program for standard JCL TSOPROGRAM is available for nonworklist JCL. | AJXIKJ |
| TSOSUBEXIT=N              | specifies whether to use a TSO submit exit to generate job cards If TSOSUBEXIT=Y, no job cards are put in the JCL. | AJXFTSOE |
**ULLQ**

ULLQ for user-defined data sets for the installation environment.

During installation, if you chose to use the runtime enablement feature, the Installation System set this value to blank, and you should not change the value. If you chose not to use the feature, the Installation System set the value to UDB.

**Note:** Changing the qualifier of the data sets might cause unpredictable results. Do not change the LLQ for the data sets.

UNLD_FREF_DATACLASS for ALTER and CHANGE MANAGER, specifies the SMS definition for the data class associated with the file reference (SYSREC) data set.

UNLD_FREF_DIRBLOCK = 250 for ALTER and CHANGE MANAGER, specifies the number of directory blocks for the file reference (SYSREC) data set.

UNLD_FREF_MGMTCLASS for ALTER and CHANGE MANAGER, specifies the SMS definition for the management class associated with the file reference (SYSREC) data set.

UNLD_FREF_PREFIX = &PREFIX..&MSSID..&WORKID8 for ALTER and CHANGE MANAGER, specifies the prefix for the name of the file reference (SYSREC) data set.

UNLD_FREF_PRIQTY = 10 for ALTER and CHANGE MANAGER, specifies the primary allocation (in cylinders) for the file reference (SYSREC) data set. Valid values are 1 through 99999.

UNLD_FREF_SECQTY = 2 for ALTER and CHANGE MANAGER, specifies the secondary allocation (in cylinders) for the file reference (SYSREC) data set. Valid values are 1 through 99999.

UNLD_FREF_STORCLASS for ALTER and CHANGE MANAGER, specifies the SMS definition for the storage class associated with the file reference (SYSREC) data set.

UNLD_FREF_UNIT = SYSDA for ALTER and CHANGE MANAGER, specifies the unit for the file reference (SYSREC) data set.

UNLD1_DATACLASS specifies the SMS definition for the data class associated with the primary unload (SYSREC) data set.

UNLD1_DATACLASS_ALT specifies the SMS definition for the data class associated with the primary unload (SYSREC) data set (used if the threshold is exceeded).
### Table 58  POF keywords (Part 48 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLD1_EXPDT</td>
<td>specifies the expiration date of the primary unload (SYSREC) data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyyy/ddd or yyyy/ddd.</td>
<td>AJXU1EX</td>
</tr>
<tr>
<td>UNLD1_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the primary unload (SYSREC) data set.</td>
<td>AJXU1MCL</td>
</tr>
<tr>
<td>UNLD1_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the primary unload (SYSREC) data set (used if the threshold is exceeded).</td>
<td>AJXMTAM</td>
</tr>
<tr>
<td>UNLD1_PREFIX= &amp;USERID..&amp;MSSID..&amp;WORKID8</td>
<td>specifies the prefix for the name of the primary unload (SYSREC) data set.</td>
<td>AJXU1PRF</td>
</tr>
<tr>
<td>UNLD1_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the primary unload (SYSREC) data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXSrips</td>
</tr>
<tr>
<td>UNLD1_RETPD</td>
<td>specifies the retention period for the primary unload (SYSREC) data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td>AJXU1RP</td>
</tr>
<tr>
<td>UNLD1_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the primary unload (SYSREC) data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.</td>
<td>AJXSrSS</td>
</tr>
<tr>
<td>UNLD1_STACK=N</td>
<td>specifies whether to stack the primary unload (SYSREC) data set on a tape with data sets of the same type.</td>
<td>AJXSTAKU</td>
</tr>
<tr>
<td>UNLD1_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set.</td>
<td>AJXU1SCL</td>
</tr>
<tr>
<td>UNLD1_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set (used if the threshold is exceeded).</td>
<td>AJXUITAS</td>
</tr>
<tr>
<td>UNLD1_THRESH=0</td>
<td>specifies the maximum anticipated size for the primary unload (SYSREC) data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td>AJXU1THX</td>
</tr>
<tr>
<td>UNLD1_UNIT=SYSDA</td>
<td>specifies the unit for the primary unload (SYSREC) data set.</td>
<td>AJXSrUNI</td>
</tr>
<tr>
<td>Keyword and default value</td>
<td>Description</td>
<td>SLIB variable</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>UNLD1_UNIT_ALT</td>
<td>specifies the alternate unit name for the primary unload (SYSREC) data set (used if the threshold value is exceeded)</td>
<td>AJXU1ALU</td>
</tr>
<tr>
<td></td>
<td>The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords.</td>
<td></td>
</tr>
<tr>
<td>UNLD2_DATACLASS</td>
<td>specifies the SMS definition for the data class associated with the backup unload (SYSREC) data set</td>
<td>AJXU2DCL</td>
</tr>
<tr>
<td>UNLD2_DATACLASS_ALT</td>
<td>specifies the SMS definition for the data class associated with the backup unload (SYSREC) data set (used if the threshold is exceeded)</td>
<td>AJXU2TAD</td>
</tr>
<tr>
<td>UNLD2_EXPDT</td>
<td>specifies the expiration date of the backup unload (SYSREC) data set on tape</td>
<td>AJXU2EX</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. The valid values are yyyyddd or yyyy/ddd.</td>
<td></td>
</tr>
<tr>
<td>UNLD2_MGMTCLASS</td>
<td>specifies the SMS definition for the management class associated with the backup unload (SYSREC) data set</td>
<td>AJXU2MCL</td>
</tr>
<tr>
<td>UNLD2_MGMTCLASS_ALT</td>
<td>specifies the SMS definition for the management class associated with the backup unload (SYSREC) data set (used if the threshold is exceeded)</td>
<td>AJXU2TAM</td>
</tr>
<tr>
<td>UNLD2_PREFIX=&amp;USERID..&amp;MSSID..&amp;WORKID8</td>
<td>specifies the prefix for the name of the backup unload (SYSREC) data set</td>
<td>AJXU2PRF</td>
</tr>
<tr>
<td>UNLD2_PRIQTY=10</td>
<td>specifies the primary allocation (in cylinders) for the backup unload (SYSREC) data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXU2PS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>UNLD2_RETPD</td>
<td>specifies the retention period for the backup unload (SYSREC) data set on tape</td>
<td>AJXU2RP</td>
</tr>
<tr>
<td></td>
<td>A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td></td>
</tr>
<tr>
<td>UNLD2_SECQTY=2</td>
<td>specifies the secondary allocation (in cylinders) for the backup unload (SYSREC) data set if DATASETSIZING=N or if an error in sizing occurs</td>
<td>AJXU2SS</td>
</tr>
<tr>
<td></td>
<td>Valid values are 1 through 99999.</td>
<td></td>
</tr>
<tr>
<td>UNLD2_STACK=N</td>
<td>specifies whether to stack the backup unload (SYSREC) data set on a tape with data sets of the same type</td>
<td>AJXU2STK</td>
</tr>
<tr>
<td>UNLD2_STORCLASS</td>
<td>specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set</td>
<td>AJXU2SCL</td>
</tr>
<tr>
<td>UNLD2_STORCLASS_ALT</td>
<td>specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set (used if the threshold is exceeded)</td>
<td>AJXU2TAS</td>
</tr>
</tbody>
</table>
### Table 58  POF keywords (Part 50 of 52)

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLD2_THRESH=0</td>
<td>specifies the maximum anticipated size for the backup unload (SYSREC) data set, in cylinders. If this value is exceeded, the alternate unit and the alternate SMS keywords will be used. Zero indicates that a threshold is not specified for the unit. If zero is specified, an alternate unit and the alternate SMS keywords will not be used.</td>
<td>AJXU2THX</td>
</tr>
<tr>
<td>UNLD2_UNIT=SYSDA</td>
<td>specifies the unit for the backup unload (SYSREC) data set</td>
<td>AJXU2UNI</td>
</tr>
<tr>
<td>UNLD2_UNIT_ALT</td>
<td>specifies the alternate unit name for the backup unload (SYSREC) data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords.</td>
<td>AJXU2ALU</td>
</tr>
<tr>
<td>UNLD3_DATACLASS</td>
<td>for the BMC UNLOAD PLUS utility in the Database Administration solution, specifies the SMS definition for the data class associated with the ROWID SYSREC data set</td>
<td>none</td>
</tr>
<tr>
<td>UNLD3_EXPDT</td>
<td>for the BMC UNLOAD PLUS utility in the Database Administration solution, specifies the expiration date of the ROWID SYSREC data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.</td>
<td>none</td>
</tr>
<tr>
<td>UNLD3_MGMTCLASS</td>
<td>for the BMC UNLOAD PLUS utility in the Database Administration solution, specifies the SMS definition for the management class associated with the ROWID SYSREC data set</td>
<td>none</td>
</tr>
<tr>
<td>UNLD3_PREFIX=&amp;USERID..&amp;MSSID..&amp;WORKID8</td>
<td>for the BMC UNLOAD PLUS utility in the Database Administration solution, specifies the prefix for the name of the ROWID SYSREC data set</td>
<td>none</td>
</tr>
<tr>
<td>UNLD3_RETPD</td>
<td>for the BMC UNLOAD PLUS utility in the Database Administration solution, specifies the retention period for the ROWID SYSREC data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.</td>
<td>none</td>
</tr>
<tr>
<td>UNLD3_STORCLASS</td>
<td>for the BMC UNLOAD PLUS utility in the Database Administration solution, specifies the SMS definition for the storage class associated with the ROWID SYSREC data set</td>
<td>none</td>
</tr>
<tr>
<td>UNLD3_UNIT=SYSDA</td>
<td>for the BMC UNLOAD PLUS utility in the Database Administration solution, specifies the unit for the ROWID SYSREC data set</td>
<td>none</td>
</tr>
</tbody>
</table>
**Table 58  POF keywords (Part 51 of 52)**

<table>
<thead>
<tr>
<th>Keyword and default value</th>
<th>Description</th>
<th>SLIB variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLD4_DATACLASS</td>
<td>for the LOB DATA MOVER program in the Database Administration solution, specifies the SMS definition for the data class associated with the large object (LOB) SYSREC data set</td>
<td>none</td>
</tr>
<tr>
<td>UNLD4_MGMTCLASS</td>
<td>for the LOB DATA MOVER program in the Database Administration solution, specifies the SMS definition for the management class associated with the LOB SYSREC data set</td>
<td>none</td>
</tr>
<tr>
<td>UNLD4_PREFIX= &amp;USERID..&amp;MSSID..&amp;WORKID8</td>
<td>for the LOB DATA MOVER program in the Database Administration solution, specifies the prefix for the name of the LOB SYSREC data set</td>
<td>none</td>
</tr>
<tr>
<td>UNLD4_STORCLASS</td>
<td>for the LOB DATA MOVER program in the Database Administration solution, specifies the SMS definition for the storage class associated with the LOB SYSREC data set</td>
<td>none</td>
</tr>
<tr>
<td>UNLD4_UNIT=SYSDA</td>
<td>for the LOB DATA MOVER program in the Database Administration solution, specifies the unit for the LOB SYSREC data set</td>
<td>none</td>
</tr>
<tr>
<td>UNLOAD+_LOAD</td>
<td>specifies the name of the LINK library for the BMC UNLOAD PLUS utility</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The BMC_UNLOAD_LOAD AJXPOFIN keyword replaces this keyword. If both BMC_UNLOAD_LOAD and UNLOAD+_LOAD are included in the POF, the components use the value that is specified for UNLOAD+_LOAD.</td>
<td></td>
</tr>
<tr>
<td>UNLOADDOPT</td>
<td>specifies the name of the installation options module for the BMC UNLOAD PLUS utility</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>The BMC_UNLOAD_OPTS AJXPOFIN keyword replaces this keyword. If both BMC_UNLOAD_OPTS and UNLOADDOPT are included in the POF, the components use the value that is specified for UNLOADDOPT.</td>
<td></td>
</tr>
<tr>
<td>USER_VAR1_CHAR</td>
<td>specifies a user-defined character variable</td>
<td>AJXUVR1</td>
</tr>
<tr>
<td>USER_VAR2_CHAR</td>
<td></td>
<td>AJXUVR2</td>
</tr>
<tr>
<td>USER_VAR3_CHAR</td>
<td></td>
<td>AJXUVR3</td>
</tr>
<tr>
<td>USER_VAR4_CHAR</td>
<td></td>
<td>AJXUVR4</td>
</tr>
<tr>
<td>USER_VAR5_CHAR</td>
<td></td>
<td>AJXUVR5</td>
</tr>
</tbody>
</table>
You can use symbolic variables in the installation options module, in the product options file (POF), and on several of the options panels. The symbolic variables are used in job cards and data set names. Table 59 lists symbolic variables that are used in all BMC Administrative products. Some of these variables are not applicable to CATALOG MANAGER. The table also lists the related skeleton library (SLIB) (or ISPF) variable and the corresponding OUTPUT and TEMPLATE descriptor variables.

On product panels, an ampersand (&) must precede the symbolic variable. In the JCL of the installation options module, two ampersands (&&) must precede the symbolic variable. For more information about the use of symbolic variables, see the installation guide.
### Symbolic variables

#### Table 59  Symbolic variables (Part 1 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJXODS44</td>
<td>1</td>
<td>Y, N</td>
<td>AJXDSN44</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALID</td>
<td>8</td>
<td>ALTER CATMGR DASDMGR</td>
<td>AJXJAID</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>ATTACH</td>
<td>4</td>
<td>none</td>
<td>AJXJSSID</td>
<td>ATTACH</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>BMCCP</td>
<td>2 to 4</td>
<td>none</td>
<td>AJXBMCCP</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CR</td>
<td>8</td>
<td>none</td>
<td>AJXCR</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>DA</td>
<td>2</td>
<td>DD</td>
<td>AJXYMD</td>
<td>DAY</td>
<td>DA</td>
</tr>
<tr>
<td>DAY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATE</td>
<td>6</td>
<td>YYMMDD</td>
<td>AJXYMD</td>
<td>DATE</td>
<td>DATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATEJ</td>
<td>7</td>
<td>YYYYDDD</td>
<td>AJX4YDDD</td>
<td>JDATE</td>
<td>JDATE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB</td>
<td>8</td>
<td>name of current database</td>
<td>AJXDB</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>DBNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB2V2</td>
<td>2</td>
<td>version of DB2</td>
<td>AJXDB2V2</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbolic variable and description</td>
<td>Size</td>
<td>Value</td>
<td>Related SLIB variable</td>
<td>OUTPUT descriptor variable</td>
<td>TEMPLATE descriptor variable</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------</td>
<td>-------------</td>
<td>------------------------</td>
<td>---------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>DB2V3</td>
<td>3</td>
<td>version of DB2</td>
<td>AJXDB2V3</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>DDD</td>
<td>3</td>
<td>DDD</td>
<td>AJXYYDDD</td>
<td>JDAY</td>
<td>JDAY</td>
</tr>
<tr>
<td>Julian day</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDNAME</td>
<td>8</td>
<td>none</td>
<td>AJXJDDN</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>DDname</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDOPT</td>
<td>8</td>
<td>none</td>
<td>AJXDDOPT</td>
<td>SEQ</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>name the of the installation options module for the BMC DASD MANAGER PLUS product</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDSEQ</td>
<td>4</td>
<td>none</td>
<td>AJXDDSQC</td>
<td>SEQ</td>
<td>SEQ</td>
</tr>
<tr>
<td>sequence number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSNUM</td>
<td>3</td>
<td>none</td>
<td>AJXPARTC</td>
<td>DSNUM</td>
<td>PART</td>
</tr>
<tr>
<td>partition number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DT</td>
<td>6</td>
<td>YYMMDD</td>
<td>AJXYMD</td>
<td>DATE</td>
<td>DT</td>
</tr>
<tr>
<td>system date (same format as JYMD and YMD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCMD</td>
<td>8</td>
<td>none</td>
<td>AJXFCMD</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>full command name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GDG³</td>
<td>4</td>
<td>initially 1</td>
<td>AJXGDGPC</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>generation data group (GDG) counter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRPNM</td>
<td>4</td>
<td>none</td>
<td>AJXJSSID</td>
<td>SSID</td>
<td>SSID</td>
</tr>
<tr>
<td>DB2 subsystem ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HM</td>
<td>4</td>
<td>HHMM</td>
<td>AJXHM</td>
<td>HOUR.MINUTE</td>
<td>HO.MI</td>
</tr>
<tr>
<td>time of JCL creation in hours and minutes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HMS</td>
<td>6</td>
<td>HHMMSS</td>
<td>AJXHMS</td>
<td>TIME</td>
<td>TIME</td>
</tr>
<tr>
<td>time of JCL creation in hours, minutes, and seconds</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 59  Symbolic variables (Part 3 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>HO HOUR</td>
<td>2</td>
<td>HH</td>
<td>AJXHMS</td>
<td>HOUR</td>
<td>HO</td>
</tr>
<tr>
<td>hour part of HHMMSS format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IC ICTYPE</td>
<td>1</td>
<td>L = local R = remote</td>
<td>AJXIC</td>
<td>ICTYPE</td>
<td>IC ICTYPE</td>
</tr>
<tr>
<td>image copy type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS index space name</td>
<td>8</td>
<td>none</td>
<td>AJIXSPC</td>
<td>TS</td>
<td>TS</td>
</tr>
<tr>
<td>IX IXNAME index name</td>
<td>16</td>
<td>name of current index</td>
<td>AJXIX</td>
<td>TS</td>
<td>IS</td>
</tr>
<tr>
<td>IXCR index creator name</td>
<td>8</td>
<td>creator of current index</td>
<td>AJXCR</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>IXNODE\textsuperscript{D} index node</td>
<td>22</td>
<td>none</td>
<td>AJXIXNOD</td>
<td>DB.TS</td>
<td>DB..IS</td>
</tr>
<tr>
<td>IXSPC index space name</td>
<td>8</td>
<td>none</td>
<td>AJIXSPC</td>
<td>TS</td>
<td>IS</td>
</tr>
<tr>
<td>JD Julian date shown with two-digit year</td>
<td>3</td>
<td>YYDDD</td>
<td>AJXYYDDD</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>JDATE Julian date shown with four-digit year</td>
<td>7</td>
<td>YYYYDDD</td>
<td>AJX4YDDD</td>
<td>JDATE</td>
<td>JDATE</td>
</tr>
<tr>
<td>JDAY Julian day</td>
<td>3</td>
<td>DDD</td>
<td>AJXYYDDD</td>
<td>JDAY</td>
<td>JDAY</td>
</tr>
<tr>
<td>JDDN ddbname for skeleton</td>
<td>8</td>
<td>none</td>
<td>AJXDDN</td>
<td>SEQ</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>JHMS time of work ID creation</td>
<td>6</td>
<td>HHMMSS</td>
<td>AJXHMS</td>
<td>TIME</td>
<td>TIME</td>
</tr>
</tbody>
</table>
### Table 59  Symbolic variables (Part 4 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>JIULD    Julian date of work ID creation</td>
<td>5</td>
<td>YYDDD</td>
<td>AJXJIULD</td>
<td>JDAY</td>
<td>JDAY</td>
</tr>
<tr>
<td>JOBCHAR  a single character that indicates the component for which JCL is being generated</td>
<td>1</td>
<td>A = Analysis</td>
<td>AJXFJCHR</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B = Baseline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C = Compare</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>E = Execution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>I = Import</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOBNAME  eight–character work ID or name assigned to a job</td>
<td>8</td>
<td>work ID or job name in the JOB statement</td>
<td>AJXWKID AJXJOBNM</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>JOBTYP   type of job</td>
<td>8</td>
<td>ANALYSIS EXECUTION Baseline COMPARE IMPORT CATMGR UTILITY DEFINE CAT ALTER CHGMGR DASDMGR</td>
<td>AJXJOBT</td>
<td>TYPE</td>
<td>IC</td>
</tr>
<tr>
<td>JPCOD    product code</td>
<td>3</td>
<td>ACM = CHANGE MANAGER</td>
<td>AJXJPCOD</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACT = CATALOG MANAGER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ALU = ALTER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ASU = DASD MANAGER PLUS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JS1      one-digit job sequence number</td>
<td>1</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>JS2      two-digit job sequence number</td>
<td>2</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
</tbody>
</table>
### Table 59  Symbolic variables (Part 5 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>JS4</td>
<td>4</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>four-digit job sequence number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JQID</td>
<td>8</td>
<td>work ID</td>
<td>AJXJQID</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>equal to the work ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSSID&lt;sup&gt;c&lt;/sup&gt;</td>
<td>4</td>
<td>none</td>
<td>AJXSSID</td>
<td>SSID</td>
<td>SSID</td>
</tr>
<tr>
<td>DB2 subsystem ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JU</td>
<td>7</td>
<td>YYYYDDD</td>
<td>AJX4YDDD</td>
<td>JDATE</td>
<td>JDATE</td>
</tr>
<tr>
<td>Julian date shown with four-digit year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JUL4Y</td>
<td>7</td>
<td>YYYYDDD</td>
<td>AJX4YDDD</td>
<td>JDATE</td>
<td>JDATE</td>
</tr>
<tr>
<td>Julian date</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JULIAN</td>
<td>5</td>
<td>YYDDD</td>
<td>AJXYYDDD</td>
<td>JDATE</td>
<td>JDATE</td>
</tr>
<tr>
<td>system date</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JYMD</td>
<td>6</td>
<td>YYMMDD</td>
<td>AJXJYMD</td>
<td>DATE</td>
<td>DATE</td>
</tr>
<tr>
<td>date of work ID creation (same as DATE)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDSNUM</td>
<td>3</td>
<td>none</td>
<td>AJXPARTC</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>partition number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LI</td>
<td>8</td>
<td>none</td>
<td>AJXLDEFN</td>
<td>UTIL</td>
<td>LI</td>
</tr>
<tr>
<td>LIST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>LIST</td>
</tr>
<tr>
<td>LISTDEF name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLQ</td>
<td>4</td>
<td>low-level qualifier for ISPF data sets</td>
<td>AJXLLQ</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>image copy type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCREM</td>
<td>1</td>
<td>L = local R = remote</td>
<td>AJXLR</td>
<td>TYPE</td>
<td>LOCREM</td>
</tr>
<tr>
<td>image copy type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LPART</td>
<td>3</td>
<td>none</td>
<td>AJXPARTC</td>
<td>LPART</td>
<td>PART</td>
</tr>
<tr>
<td>partition number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td>1</td>
<td>L = local R = remote</td>
<td>AJXLR</td>
<td>TYPE</td>
<td>LR</td>
</tr>
<tr>
<td>image copy type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Symbolic variables (Part 6 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBER MEMBR PDS member name</td>
<td>8</td>
<td>first seven characters of the work ID, with one of the following values appended: J = JCL W = worklist D = diagnostics C = CDL S = source (CM/PILOT) T = target (CM/PILOT) Z = other</td>
<td>AJXMEMBER AJXMEMBR</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>MI MINUTE JCL date</td>
<td>2</td>
<td>MM</td>
<td>AJXHMS</td>
<td>MINUTE</td>
<td>MI MINUTE</td>
</tr>
<tr>
<td>MMDD JCL date</td>
<td>4</td>
<td>MMDD</td>
<td>MONTH.DAY</td>
<td>MONTH.DAY</td>
<td>MONTH.DAY</td>
</tr>
<tr>
<td>MO MONTH JCL date</td>
<td>2</td>
<td>MM</td>
<td>AJXYMD</td>
<td>MONTH</td>
<td>MO MONTH</td>
</tr>
<tr>
<td>MSSID DB2 subsystem ID</td>
<td>4</td>
<td>determined at runtime from the -JCLP command in the worklist</td>
<td>AJXMISSID</td>
<td>SSID</td>
<td>SSID</td>
</tr>
<tr>
<td>OBJT OBJTYP object type</td>
<td>2</td>
<td>two-character code that indicates the object type: TS, TB, IX</td>
<td>AJXOBT</td>
<td>UTIL TYPE</td>
<td>IC</td>
</tr>
<tr>
<td>OBNAME OBNAME object name</td>
<td>27</td>
<td>none</td>
<td>AJXOBNAME</td>
<td>DB.TS</td>
<td>DB..SN</td>
</tr>
<tr>
<td>OBnod OBnod database and either table space name or index space name</td>
<td>17</td>
<td>#### (for objects that do not exist)</td>
<td>AJXOBNOD</td>
<td>DB.TS</td>
<td>DB..SN</td>
</tr>
</tbody>
</table>
Table 59  Symbolic variables (Part 7 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA  partition number</td>
<td>3</td>
<td>none</td>
<td>AJXPARTC</td>
<td>PART</td>
<td>PA</td>
</tr>
<tr>
<td>PART  four-digit partition number in which insignificant digits are suppressed</td>
<td>4</td>
<td>none</td>
<td>AJXPARTC</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>PART4  four-digit partition number in which leading zeros are not suppressed</td>
<td>4</td>
<td>none</td>
<td>AJX4PART</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>PART5  five-digit partition number in which leading zeros are not suppressed</td>
<td>5</td>
<td>none</td>
<td>AJX5PART</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>PB PRIBAC  type of copy</td>
<td>1</td>
<td>P = primary B = backup</td>
<td>AJXPB</td>
<td>TYPE</td>
<td>PB PRIBAC</td>
</tr>
<tr>
<td>PGMIR  programmer name field</td>
<td>20</td>
<td>‘&amp;&amp;JOBTYP - &amp;&amp;WKID’</td>
<td>AJXPGMIR</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>(ALTER and CHANGE MANAGER) PREFIX TSO prefix (same as ZPREFIX)</td>
<td>8</td>
<td>TSO prefix (&amp;ZUSER if NOPREFIX) or user ID</td>
<td>ZPREFIX</td>
<td>none</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>RHLQ  high-level qualifier (HLQ) for ISPF data sets</td>
<td>60</td>
<td>none</td>
<td>AJXRHLQ</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>RSEQ#  restart sequence number</td>
<td>6</td>
<td>none</td>
<td>AJXRSEQ#</td>
<td>SEQ</td>
<td>SEQ</td>
</tr>
<tr>
<td>RTYPE  object type</td>
<td>2</td>
<td>none</td>
<td>AJXOBJT</td>
<td>TYPE</td>
<td>IC</td>
</tr>
</tbody>
</table>
### Table 59  Symbolic variables (Part 8 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUNTYP type of run</td>
<td>9</td>
<td>ANALYZE EXECUTE BASELINE COMPARE IMPORT ALTER RESTART STARTOVER</td>
<td>AJXRUNTP</td>
<td>TYPE</td>
<td>IC</td>
</tr>
<tr>
<td>SC SEC SECOND second part of HHMMSS format</td>
<td>2</td>
<td>SS</td>
<td>AJXHMS</td>
<td>SEC SEC SECOND</td>
<td>SC SC SECOND</td>
</tr>
<tr>
<td>SEQ DD sequence number</td>
<td>4</td>
<td>none</td>
<td>AJXDDSQC</td>
<td>SEQ</td>
<td>SEQ</td>
</tr>
<tr>
<td>SEQ# sequence number</td>
<td>6</td>
<td>none</td>
<td>AJXSEQ#</td>
<td>SEQ</td>
<td>SEQ</td>
</tr>
<tr>
<td>SN either table space name or index space name</td>
<td>8</td>
<td>none</td>
<td>AJXSPNAM</td>
<td>TS</td>
<td>SN</td>
</tr>
<tr>
<td>SPNAME either table space name or index space name</td>
<td>8</td>
<td>none</td>
<td>AJXSPNAM</td>
<td>TS</td>
<td>SN</td>
</tr>
<tr>
<td>SQ DD sequence number</td>
<td>4</td>
<td>none</td>
<td>AJXDDSQC</td>
<td>SEQ</td>
<td>SQ</td>
</tr>
<tr>
<td>SS DB2 subsystem ID</td>
<td>4</td>
<td>none</td>
<td>AJXJSSID</td>
<td>SSID</td>
<td>SS</td>
</tr>
<tr>
<td>SSID DB2 subsystem ID (same as JSSID)</td>
<td>4</td>
<td>determined at runtime</td>
<td>AJXSSID</td>
<td>SSID</td>
<td>SSID</td>
</tr>
<tr>
<td>ST STEPN STEPNAME step name</td>
<td>8</td>
<td>none</td>
<td>AJXSTEPN</td>
<td>STEPNAME</td>
<td>STEPNAME</td>
</tr>
</tbody>
</table>
### Symbolic variables (Part 9 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEP#</td>
<td>6</td>
<td>none</td>
<td>AJXSTEPC</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>step number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSMLIB</td>
<td>46</td>
<td>name of system ISPF message library</td>
<td>AJXSYSM</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>SYSSID</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>JCL symbolic parameter</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBCR TBCRE</td>
<td>8</td>
<td>owner of current table</td>
<td>AJXTBCR AJXTBCRE</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>table creator name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBNAMd TBNAM</td>
<td>12</td>
<td>name of current table</td>
<td>AJXTBNAM</td>
<td>TS</td>
<td>TS</td>
</tr>
<tr>
<td>table name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBNODE</td>
<td>22</td>
<td>none</td>
<td>AJXTBNOD</td>
<td>DB.TS</td>
<td>DB..TS</td>
</tr>
<tr>
<td>table node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TI TIME</td>
<td>6</td>
<td>HHMMSS</td>
<td>AJXHMS</td>
<td>TIME</td>
<td>TI TIME</td>
</tr>
<tr>
<td>system time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME4</td>
<td>4</td>
<td>HHMM</td>
<td>AJXHMS</td>
<td>HOUR.MINUTE</td>
<td>HO.MI</td>
</tr>
<tr>
<td>hours and minutes of HHMMSS format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS TSNAME</td>
<td>8</td>
<td>name of current table space</td>
<td>AJXTS</td>
<td>TS</td>
<td>TS</td>
</tr>
<tr>
<td>table space name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSCR</td>
<td>8</td>
<td>none</td>
<td>AJXTSCR</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>table space creator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSIX</td>
<td>8</td>
<td>none</td>
<td>AJXSPNAM</td>
<td>TS</td>
<td>SN</td>
</tr>
<tr>
<td>either table space name or index space name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSSID DB2 subsystem ID</td>
<td>4</td>
<td>determined at runtime from the -JCLP command in the worklist</td>
<td>AJXTSSID</td>
<td>SSID</td>
<td>SSID</td>
</tr>
<tr>
<td>TU1 tape unit 1</td>
<td>4</td>
<td>none</td>
<td>AJXTU1</td>
<td>STEPNAME</td>
<td>STEPNAME</td>
</tr>
</tbody>
</table>
### Table 59  Symbolic variables (Part 10 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB descriptor variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>TU2 tape unit 2</td>
<td>4</td>
<td>none</td>
<td>AJXTU2</td>
<td>STEPNAME</td>
<td>STEPNAME</td>
</tr>
<tr>
<td>TU3 tape unit 3</td>
<td>4</td>
<td>none</td>
<td>AJXTU3</td>
<td>STEPNAME</td>
<td>STEPNAME</td>
</tr>
<tr>
<td>TYPE type of copy</td>
<td>1</td>
<td>L = local R = remote</td>
<td>AJXLR TYPE</td>
<td>IC</td>
<td></td>
</tr>
<tr>
<td>UCMD command ID</td>
<td>4</td>
<td>none</td>
<td>AJXUCMD UTIL UT</td>
<td>UT</td>
<td></td>
</tr>
<tr>
<td>UDOPT installation options file lname</td>
<td>8</td>
<td>none</td>
<td>AJXUDOPT STEPNAME STEPNAME</td>
<td>STEPNAME</td>
<td></td>
</tr>
<tr>
<td>UID TSO user ID</td>
<td>7</td>
<td>none</td>
<td>ZUSER none <strong>PREFIX</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ULLQ low-level qualifier for user-defined data sets</td>
<td>4</td>
<td>low-level qualifier for user-defined data sets</td>
<td>AJXULLQ none none</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>UP UPART three-digit partition number variable compared to AJXPARTC in which insignificant digits are suppressed</td>
<td>3</td>
<td>none</td>
<td>AJXUPART PART PART</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>USER1 user-defined</td>
<td>none</td>
<td>none</td>
<td>none none UTIL UT</td>
<td>UT</td>
<td></td>
</tr>
<tr>
<td>USER2 user-defined</td>
<td>none</td>
<td>none</td>
<td>none none UTIL UT</td>
<td>UT</td>
<td></td>
</tr>
<tr>
<td>USERID TSO user ID</td>
<td>7</td>
<td>TSO user ID ZUSER USERID <strong>PREFIX</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 59  Symbolic variables (Part 11 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>UT UTID UTIL UTILID</td>
<td>16</td>
<td>none</td>
<td>AJXUTID</td>
<td>UTIL UTILID UTIL UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>UTILITY ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTILPFX</td>
<td>8</td>
<td>none</td>
<td>AJXUTID</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>first eight bytes of utility ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTILSFX</td>
<td>8</td>
<td>none</td>
<td>AJXUTID</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>last eight bytes of utility ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UVR1 UVR2 UVR3 UVR4 UVR5</td>
<td>8</td>
<td>user-defined variable or ######## (if value is blank)</td>
<td>AJXUVR1 AJXUVR2 AJXUVR3 AJXUVR4 AJXUVR5</td>
<td>user-defined</td>
<td>user-defined</td>
</tr>
<tr>
<td>user-defined character variable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VCAT</td>
<td>8</td>
<td>none</td>
<td>AJXVCAT</td>
<td>DB DB</td>
<td>DB</td>
</tr>
<tr>
<td>VCAT name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WKID</td>
<td>8</td>
<td>name of the current work ID in use</td>
<td>AJXJQID</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>name of the work ID for the BMC DASD MANAGER PLUS product</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WKOWN WKOWNER</td>
<td>8</td>
<td>owner of the current work ID in use</td>
<td>AJXWKOWN JOBNAME STEPNAME</td>
<td>JOBNAME STEPNAME</td>
<td></td>
</tr>
<tr>
<td>work ID owner</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WORKID(^\text{a}) work ID name</td>
<td>18</td>
<td>the name of the current work ID in use or, for Compare only, the outbound migrate profile name specified that is for use If the work ID name contains characters that are invalid for use in data set names, the work ID will be truncated at the first invalid character.</td>
<td>AJXJQID UTIL</td>
<td>UT</td>
<td>UT</td>
</tr>
</tbody>
</table>
### Table 59  Symbolic variables (Part 12 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
</table>
| WORKID8  
first eight characters of the work ID name                          | 8    | If the work ID name contains characters that are invalid for use in data set names, the work ID will be truncated at the first invalid character. | AJXWKID                     | UTIL                        | UT                           |
| YE  
YEAR  
four-digit year from Julian date                                    | 4    | YYYY  | AJX4YDDD                | YEAR                        | YE                           |
| YY  
two-digit year from Julian date                                      | 2    | YY    | AJXYYDDD                | none                        | none                         |
| YMD  
date of JCL creation (same as DATE and JYMD)                         | 6    | YYMMDD | AJXYMD                  | DATE                        | DATE                         |
| YYDDD  
Julian date of JCL creation (same as JULIAN)                          | 5    | YYDDD  | AJXYYDDD                | JDATE                       | JDATE                         |
| YYYYDDD  
Julian date shown with four-digit year                                 | 7    | YYYYDDD | AJX4YDDD                | JDATE                       | JDATE                         |
| ZACCTNUM  
user’s account number for jobs that are generated by the product     | 40   | If the replacement value is not known or does not fit in the space provided, question marks (?) are substituted. | ZACCTNUM                  | USERID                      | JOBNAME                      |
| ZPREFIX$^1$  
TSO prefix                                                           | 8    | none   | ZPREFIX                  | USERID                      | **PREFIX**                   |

---

$^1$ TSO prefix
Table 59  Symbolic variables (Part 13 of 13)

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="#">ZSYSID</a> ISPF system variable</td>
<td></td>
<td>system name</td>
<td>ZSYSID</td>
<td>ATTACH</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>ZUSER user ID</td>
<td>8</td>
<td>none</td>
<td>ZUSER</td>
<td>USERID</td>
<td><strong>PREFIX</strong></td>
</tr>
</tbody>
</table>

a  This variable is incremented each time any image copy is taken for a specific table space. Consider the following items if you use the GDG variable:
- JCL Generation does not verify that the GDG base definitions already exist.
- No GDG numbers are built for invalid or incomplete table space names. JCL Generation builds the &AJXDB, &AJXTS, and &OBNOD variables by parsing the utility commands in the worklist. An alter-type worklist might contain incomplete table space names for implicit table spaces because these names will not be known until the worklist is executed.

b  This value truncates after eight characters when used by JCL Generation.

c  This value is determined at runtime (same as SSID).

d  This value truncates after eight characters when used by JCL Generation.

e  This value truncates after eight characters when used by JCL Generation. For the Compare component of CHANGE MANAGER, this variable indicates the outbound migrate profile name.

f  Leave this variable blank for NOPREFIX (same as PREFIX).
This appendix presents the following topics:

CATALOG MANAGER commands .......................................................... 455
DB2 action commands ........................................................................ 459
Utility commands ................................................................................ 461
Utility list commands .......................................................................... 464
Statistics commands ........................................................................... 464
List commands .................................................................................... 465
User commands ................................................................................... 469
Data browsing and editing commands .................................................. 470
  Command-line commands .................................................................. 470
  Line commands ................................................................................ 473

CATALOG MANAGER commands

CATALOG MANAGER commands are contained in a separately assembled and linked module. This module might have been modified at your installation. You can display a list of the commands in your command module by entering the COMMANDS (CMD) command. From this list, you can select the Help panel of each command by entering S on the line next to the command name. Table 60 lists the commands.

Table 60  CATALOG MANAGER commands (part 1 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2WL</td>
<td>when used from the SQL_Table list, creates a worklist from the selected SQL member</td>
</tr>
<tr>
<td>ANALYZE</td>
<td>displays a SELECT, INSERT, UPDATE, DELETE, or DECLARE CURSOR SQL statement from the CATALOG MANAGER SQL_Table</td>
</tr>
<tr>
<td></td>
<td>You can invoke the BMC SQL Explorer Analysis function to analyze the SQL.</td>
</tr>
</tbody>
</table>
## Table 60  CATALOG MANAGER commands (part 2 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| APPLY        | creates SQL statements by using a model with host variables into which CATALOG MANAGER will substitute values from the DB2 catalog.  
               | The name of the host variable should correspond to the column name of the value to be substituted.                                           |
| AUDIT        | displays rows from the CATALOG MANAGER Audit Log table                                                                                     |
| BATCH        | generates JCL for a CATALOG MANAGER job to run in batch mode. Use BATCH when lengthy processing time makes running the job online undesirable.  |
| BROWSE       | invokes the CATALOG MANAGER data browsing feature to browse data in the selected table or view.                                            |
| CANCEL       | terminates the current display without saving any changes that might have been made                                                        |
| CASCADE      | displays the privileges that the REVOKE command would revoke for the specified user                                                        |
| CATALOGHELP  | displays information about columns in the DB2 catalog tables                                                                                |
| CLIPBOARD    | displays the Confirm SQL panel for the contents of the CATALOG MANAGER clipboard. On this panel, you can edit, save, and execute the SQL from the clipboard.  
<pre><code>           | CATALOG MANAGER saves changes in the CATALOG MANAGER SQL_Table, not in the clipboard.                                                    |
</code></pre>
<p>| CLIST        | allows CATALOG MANAGER to execute a CLIST that is available but is not in the commands table                                               |
| CMD          | see COMMANDS                                                                                                                             |
| COMMANDS     | lists the valid CATALOG MANAGER commands for the current list type. For example, different commands are listed for tables and columns.      |
| COPYAUTHS    | generates SQL that will copy authorizations from a source user to a target user and will grant authorizations from a source object to a target object |
| CUSTOMIZE    | allows you to customize the CATALOG MANAGER Primary Menu panel to include a subset of the options                                           |
| D            | displays the catalog row for the selected object                                                                                          |
| DCL          | generates GRANTs for explicit privileges that are held on an object or by a user                                                          |
| DDL          | displays the DDL for the selected object                                                                                                  |
| DES          | displays a subset of the description that is provided by the DESCRIBE command for tables, databases, and DBRM packages                     |
| DESCRIBE     | displays catalog information about the selected object. In some instances, the information might come from more than one catalog table; for example, a DESCRIBE command on a database will show the plans that are associated with the database. |
| DOPTS        | displays the installation options for this session of CATALOG MANAGER                                                                     |</p>
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DROPRECOVERY</td>
<td>displays a list of objects that CATALOG MANAGER dropped with Recovery On specified, and allows you to select an object to recover</td>
</tr>
<tr>
<td>EDIT</td>
<td>invokes the CATALOG MANAGER data editing feature to edit data in the selected table or view</td>
</tr>
<tr>
<td>ENVIRONMENT</td>
<td>displays some of the CATALOG MANAGER variables</td>
</tr>
<tr>
<td>EXPLAIN</td>
<td>displays rows from the ownerName.PLAN_TABLE for the object</td>
</tr>
<tr>
<td>GET</td>
<td>when used with the DESCRIBE command, displays a SELECT, INSERT, UPDATE, DELETE, or DECLARE CURSOR SQL statement that is stored in a DBRM or a package</td>
</tr>
<tr>
<td>HC</td>
<td>displays a panel to build, edit, and submit a job to print the CATALOG MANAGER PRINT data set while CATALOG MANAGER is running</td>
</tr>
<tr>
<td>HDDL</td>
<td>generates hierarchical DDL for the selected object</td>
</tr>
<tr>
<td>HDESCRIBE</td>
<td>generates a hierarchical describe for the selected object and routes the output list to your terminal</td>
</tr>
<tr>
<td>HGRANT</td>
<td>generates SQL GRANT statements to show the privileges that are held on the object for which the command was entered and for any dependent objects</td>
</tr>
<tr>
<td>HPRINT</td>
<td>generates a hierarchical DESCRIBE command for the selected object and routes the output list to your PRINT data set</td>
</tr>
<tr>
<td>IMPORT</td>
<td>copies SQL from a PDS into the CATALOG MANAGER SQL_TABLE</td>
</tr>
<tr>
<td>JOIN</td>
<td>builds an SQL SELECT statement template for multiple tables</td>
</tr>
<tr>
<td>LEDIT</td>
<td>allows the current list to be edited, saved, or executed as a set of SQL statements</td>
</tr>
<tr>
<td>LOGRBA</td>
<td>displays the current DB2 log RBA and the associated time stamp</td>
</tr>
<tr>
<td>MAINTAIN</td>
<td>displays the Log Maintenance Menu, on which you can browse or purge the Session Log, DDL Audit Log, and Drop Recovery Log</td>
</tr>
<tr>
<td>MDDL</td>
<td>generates DDL for all objects in the list in one stream</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>displays the CATALOG MANAGER options panels</td>
</tr>
</tbody>
</table>
## Table 60  CATALOG MANAGER commands (part 4 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDER</td>
<td>allows you to specify the order in which columns will be displayed from left to right across a list</td>
</tr>
<tr>
<td>PACKIT</td>
<td><em>(DB2 Version 9 or earlier)</em> converts a plan to use only packages</td>
</tr>
<tr>
<td></td>
<td>A PACKIT command generates a series of DSN commands, first to bind each DBRM referenced by the plan into a package, and then to bind the plan again by using the packages instead of the original DBRMs.</td>
</tr>
<tr>
<td>PRINT</td>
<td>when issued on the <strong>Command</strong> line, prints the current list</td>
</tr>
<tr>
<td></td>
<td>When issued on a list, a DESCRIBE command is executed and the output is routed to the PRINT data set.</td>
</tr>
<tr>
<td></td>
<td>The PRINT command opens the data set for output the first time that you issue the command in each session. Additional print output is appended to the data set until you issue the HC command to submit the data set for printing or the PRINT CLOSE command to close and deallocate the data set.</td>
</tr>
<tr>
<td>PRO</td>
<td>when issued on a list, displays the Profiles List panel, which lists all of the customized session profiles for CATALOG MANAGER</td>
</tr>
<tr>
<td>PROFILE</td>
<td>When issued on the <strong>Command</strong> line of the Utility Selections panel, PROFILE displays a list of user and site profiles for the selected utility type or types.</td>
</tr>
<tr>
<td>PROFILES</td>
<td></td>
</tr>
<tr>
<td>REFRESH</td>
<td>when issued on the <strong>Command</strong> line, rebuilds the current list to include any changes that have occurred since the list was originally built</td>
</tr>
<tr>
<td>RESET</td>
<td>removes values and other input information from the list line (such as the information that results from running the COUNT command).</td>
</tr>
<tr>
<td>RESPONSES</td>
<td>restores the results of list line commands after using RESET to remove them</td>
</tr>
<tr>
<td>S</td>
<td>displays the information from the DB2 catalog for the selected row</td>
</tr>
<tr>
<td>SEARCH</td>
<td>discards all current lists and creates a new list based on the search criteria that you specify</td>
</tr>
<tr>
<td>SESSION</td>
<td>displays the current Session Log records</td>
</tr>
<tr>
<td>SET</td>
<td>allows you to change the current SQLID, several CATALOG MANAGER switches, the rules for executing Type 2 CONNECT statements, and the session profile in use</td>
</tr>
<tr>
<td>SORT</td>
<td>allows you to sort the list by any displayed column</td>
</tr>
<tr>
<td>SQL</td>
<td>produces a list of saved SQL members</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>displays records from the Session Log of actions taken</td>
</tr>
<tr>
<td>TAILOR</td>
<td>displays a command list that you can tailor for a specific profile</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>converts a DB2 plan or package contoken to date and time</td>
</tr>
<tr>
<td>X</td>
<td>excludes objects from being processed by a <strong>Command</strong> line command with the keyword ALL or by a line command</td>
</tr>
</tbody>
</table>
DB2 action commands

CATALOG MANAGER generates several DB2 commands. To execute a command, type the command in the **Cmd** column next to an appropriate item on a list. Table 61 lists the DB2 action commands.

**Table 61  DB2 action commands (part 1 of 3)**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVATE</td>
<td>for a native SQL procedure, generates an SQL ALTER PROCEDURE ACTIVATE statement</td>
</tr>
<tr>
<td></td>
<td>ACTIVATE is valid only for stored procedures in which ORIGIN is N.</td>
</tr>
<tr>
<td>ALTER</td>
<td>generates an SQL ALTER statement for the object</td>
</tr>
<tr>
<td>BIND</td>
<td>displays the Bind panels, from which you can perform the following tasks:</td>
</tr>
<tr>
<td></td>
<td>- use the Explain option to indicate whether access path information is saved in <em>ownerName</em>.PLAN_TABLE</td>
</tr>
<tr>
<td></td>
<td>- input options to bind plans, packages, or DBRMs</td>
</tr>
<tr>
<td>BINDCOPY</td>
<td>binds all packages in a collection ID (CI) into a different CI, or binds all packages in a list into a single CI</td>
</tr>
<tr>
<td>BINDDEPLOY</td>
<td>displays the Bind Deploy Package panel, from which you can specify the locations to which you want to generate BIND commands</td>
</tr>
<tr>
<td></td>
<td>If the locations or wildcard pattern that you specify do not match any locations in your SYSIBM.LOCATIONS table, the product does not generate any BIND DEPLOY commands. However, if the locations or wildcard pattern match more than one location, the product groups the commands for the selected packages by location name.</td>
</tr>
<tr>
<td></td>
<td>For a native SQL procedure, the command generates a DSN BIND PACKAGE DEPLOY statement.</td>
</tr>
<tr>
<td></td>
<td>BINDDEPLOY is valid only for packages in which TYPE is N.</td>
</tr>
<tr>
<td>COMMENT</td>
<td>displays a panel to generate SQL to create or replace a comment on an alias, table, distinct type, function, index, procedure, trigger, or view</td>
</tr>
<tr>
<td>CONNECT</td>
<td>enables you to remain in your current CATALOG MANAGER session and access another DB2 subsystem on the same or another z/OS system</td>
</tr>
<tr>
<td>COUNT</td>
<td>executes an SQL COUNT(*) statement for an object and displays the number of rows in the object</td>
</tr>
<tr>
<td>CREATE</td>
<td>displays a panel from which you can input options to create an object</td>
</tr>
<tr>
<td></td>
<td>If entered on a list line, CREATE works as a CREATE LIKE of the object.</td>
</tr>
<tr>
<td>DB2COMMAND</td>
<td>lists the DB2 commands that have been saved</td>
</tr>
<tr>
<td></td>
<td>Enter <strong>DB2COMMAND PROMPT</strong> to display the same panel as the DB2 Commands action from the CATALOG MANAGER Primary Menu.</td>
</tr>
</tbody>
</table>
## Table 61  DB2 action commands (part 2 of 3)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCLGEN</td>
<td>displays panels from which you can input options to create DSN DCLGEN commands</td>
</tr>
<tr>
<td>DELETE</td>
<td>builds an SQL DELETE template for a table or view, and invokes the ISPF editor to allow you to customize the template</td>
</tr>
<tr>
<td>DIS</td>
<td>displays the status of the selected object</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>displays the status of the selected object</td>
</tr>
<tr>
<td>DISDATABASE</td>
<td>displays the status of the selected database</td>
</tr>
<tr>
<td>DISTHREAD</td>
<td>displays active threads for the current subsystem</td>
</tr>
<tr>
<td>DROP</td>
<td>generates an SQL DROP statement for the specified object and displays the Confirm SQL panel. From this panel, you can select actions and options, including the drop recovery option. When you issue the DROP command on the Pending DDL List panel, CATALOG MANAGER generates the ALTER TABLESPACE DROP PENDING CHANGES statement.</td>
</tr>
<tr>
<td>DSNZPARM</td>
<td>executes the IBM DSNWZP stored procedure and formats the information for display</td>
</tr>
<tr>
<td>EXCHANGE</td>
<td>exchanges data between two tables, one of which must be a clone of the other table</td>
</tr>
<tr>
<td>FREE</td>
<td>generates SQL to free active versions of plans or packages</td>
</tr>
<tr>
<td>FREE ALL ALL</td>
<td>generates SQL to free all plans or packages in a list and all versions of the plans or packages. FREE ALL ALL specifies the DB2 PLANMGMTSCOPE(ALL) clause.</td>
</tr>
<tr>
<td>FREE INACT</td>
<td>generates SQL to free inactive versions of plans or packages in a list. FREE INACT or FREE INACTIVE specifies the DB2 PLANMGMTSCOPE(INACTIVE) clause.</td>
</tr>
<tr>
<td>FREE INACTIVE</td>
<td></td>
</tr>
<tr>
<td>GRANT</td>
<td>generates SQL to grant privileges</td>
</tr>
<tr>
<td>INSERT</td>
<td>builds an SQL INSERT statement template for a table or view, and invokes the ISPF editor to allow you to customize the statement</td>
</tr>
<tr>
<td>LABEL</td>
<td>generates an SQL LABEL command for the selected object</td>
</tr>
<tr>
<td>QCONNECT</td>
<td>displays the Connections List panel, which lists saved connections from the current SSID. If no saved connections exist, QCONNECT displays the Change Access panel, from which you can type parameters for a connection.</td>
</tr>
</tbody>
</table>
| REBIND          | displays the Rebind panels, from which you can perform the following tasks:  
  - use the Explain option to indicate whether access path information is saved in `ownerName.PLAN_TABLE`  
  - input options to rebind plans, packages, or DBRMs                                                                                           |
Utility commands

CATALOG MANAGER generates and submits a JCL file for several BMC and DB2 utilities. To execute a utility, type the command in the Cmd column next to an appropriate item on a list. Table 62 lists the utility commands.

Table 62 Utility commands (part 1 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCCHECK</td>
<td>displays panels from which you can input options and generate BMC CHECK PLUS jobs with the CHECK DATA command</td>
</tr>
<tr>
<td>BMCCHECK INDEX</td>
<td>displays panels from which you can input options and generate BMC CHECK PLUS jobs with the CHECK INDEX command</td>
</tr>
<tr>
<td>BMCCHECK IX</td>
<td></td>
</tr>
</tbody>
</table>

Table 61 DB2 action commands (part 3 of 3)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REFRESH</td>
<td>when issued on a list line of a materialized query table (MQT), REFRESH integer refreshes the data in the MQT. integer represents the number that is associated with the SQL statement in the QUERYNO clause.</td>
</tr>
<tr>
<td>REGENERATE</td>
<td>regenerates the data in a view or an index. REGENERATE also generates an ALTER PROCEDURE REGENERATE VERSION statement for a native SQL procedure. REGENERATE is valid only for stored procedures in which ORIGIN is N.</td>
</tr>
<tr>
<td>RENAME</td>
<td>displays a panel to input a new table, index, or column name and generate RENAME SQL</td>
</tr>
<tr>
<td>REVOKE</td>
<td>displays panels from which you can input options to revoke privileges</td>
</tr>
<tr>
<td>SEE</td>
<td>allows you to modify DB2 special registers</td>
</tr>
<tr>
<td>SELECT</td>
<td>generates an SQL SELECT statement for the specified table or view, and displays a confirmation panel.</td>
</tr>
<tr>
<td>START</td>
<td>executes the DB2 START DATABASE command. From a table space list, START CLONE executes the START DATABASE command for a clone table.</td>
</tr>
<tr>
<td>STOP</td>
<td>executes the DB2 STOP DATABASE command. From a table space list, STOP CLONE executes the STOP DATABASE command for a clone table.</td>
</tr>
<tr>
<td>TRUNCATE</td>
<td>deletes all rows for a base or global temporary table</td>
</tr>
<tr>
<td>UPDATE</td>
<td>builds an SQL UPDATE statement template for a table or view, and invokes the ISPF editor for you to customize the statement</td>
</tr>
</tbody>
</table>
## Table 62 Utility commands (part 2 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCCCHTS</td>
<td>displays panels from which you can input options and generate BMC CHECK</td>
</tr>
<tr>
<td></td>
<td>PLUS jobs with the CHECK TABLESPACE command</td>
</tr>
<tr>
<td>BMCCOPY</td>
<td>displays panels from which you can input options and generate BMC COPY</td>
</tr>
<tr>
<td>BMCCOPY INDEX</td>
<td>PLUS jobs</td>
</tr>
<tr>
<td>BMCCOPY IX</td>
<td></td>
</tr>
<tr>
<td>BMCEXPLORE</td>
<td>calls the BMC SQL Explorer interface</td>
</tr>
<tr>
<td>BMCLOAD</td>
<td>displays panels from which you can input options and generate BMC LOADPLUS</td>
</tr>
<tr>
<td>BMCREBUILD</td>
<td>displays panels from which you can input options and generate BMC RECOVER</td>
</tr>
<tr>
<td>BMCREBUILD INDEX</td>
<td>PLUS jobs with the REBUILD INDEX command</td>
</tr>
<tr>
<td>BMCREBUILD IX</td>
<td></td>
</tr>
<tr>
<td>BMCRECOVER</td>
<td>displays panels from which you can input options and generate BMC RECOVER</td>
</tr>
<tr>
<td>BMCRECOVER INDEX</td>
<td>PLUS jobs</td>
</tr>
<tr>
<td>BMCRECOVER IX</td>
<td></td>
</tr>
<tr>
<td>BMCREORG</td>
<td>displays panels from which you can input options and generate BMC REORG</td>
</tr>
<tr>
<td>BMCREORG INDEX</td>
<td>PLUS jobs</td>
</tr>
<tr>
<td>BMCREORG IX</td>
<td></td>
</tr>
<tr>
<td>BMCSTATS</td>
<td>displays panels from which you can input options and generate BMCSTATS</td>
</tr>
<tr>
<td>BMCSTATS INDEX</td>
<td>jobs</td>
</tr>
<tr>
<td>BMCSTATS IX</td>
<td></td>
</tr>
<tr>
<td>BMCUHIST</td>
<td>displays information from the BMC Utility History table</td>
</tr>
<tr>
<td>BMCUNLOAD</td>
<td>displays panels from which you can input options and generate BMC UNLOAD</td>
</tr>
<tr>
<td></td>
<td>PLUS jobs</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>lists BMC utilities that have not completed or have not been terminated</td>
</tr>
<tr>
<td></td>
<td>From this list, you can terminate a utility.</td>
</tr>
<tr>
<td>CHECK CHKD</td>
<td>displays panels from which you can input options and generate IBM CHECK</td>
</tr>
<tr>
<td></td>
<td>DATA jobs</td>
</tr>
<tr>
<td>CHECK INDEX</td>
<td>displays panels from which you can input options and generate IBM CHECK</td>
</tr>
<tr>
<td>CHECK IX CHKI</td>
<td>INDEX jobs</td>
</tr>
<tr>
<td>COPY COPY INDEX</td>
<td>displays panels from which you can input options and generate IBM COPY</td>
</tr>
<tr>
<td>COPY IX</td>
<td>jobs</td>
</tr>
<tr>
<td>COPYTOCOPY COPYTOCOPY INDEX</td>
<td>displays panels from which you can input options and generate IBM COPYTOCOPY jobs</td>
</tr>
<tr>
<td>COPYTOCOPY IX</td>
<td></td>
</tr>
<tr>
<td>DISUTILITY</td>
<td>displays IBM utility information</td>
</tr>
<tr>
<td>DSN1COMP</td>
<td>displays panels from which you can input options for the IBM DSN1COMP utility</td>
</tr>
<tr>
<td>DSN1COPY</td>
<td>displays panels from which you can input options for the IBM DSN1COPY utility</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EXCEPTIONS</td>
<td>invokes the BMC DASD MANAGER PLUS product to report exceptions on the object</td>
</tr>
<tr>
<td>EXEC</td>
<td>displays panels from which you can input options for the IBM EXEC SQL utility</td>
</tr>
<tr>
<td>LOAD</td>
<td>displays panels from which you can input options for the IBM LOAD utility</td>
</tr>
<tr>
<td>MERGECOPY</td>
<td>displays panels from which you can input options for the IBM MERGECOPY utility</td>
</tr>
<tr>
<td>MODIFY</td>
<td>displays panels from which you can input options for the IBM MODIFY utility</td>
</tr>
<tr>
<td>MODIFYRECOVERY</td>
<td>displays panels from which you can input options for the IBM MODIFY utility with the RECOVERY option</td>
</tr>
<tr>
<td>MODIFYSTATISTICS</td>
<td>displays panels from which you can input options for the IBM MODIFY STATISTICS utility</td>
</tr>
<tr>
<td>QUIESCE</td>
<td>displays panels from which you can input options for the IBM QUIESCE utility</td>
</tr>
<tr>
<td>REBUILD</td>
<td>displays panels from which you can input options for the IBM REBUILD INDEX utility</td>
</tr>
<tr>
<td>REBUILD INDEX</td>
<td>displays panels from which you can input options for the IBM REBUILD INDEX utility</td>
</tr>
<tr>
<td>RECOVER</td>
<td>displays panels from which you can input options for the IBM RECOVER utility</td>
</tr>
<tr>
<td>RECOVER INDEX</td>
<td>displays panels from which you can input options for the IBM RECOVER INDEX utility</td>
</tr>
<tr>
<td>REORG</td>
<td>displays panels from which you can input options for the IBM REORG TABLESPACE utility</td>
</tr>
<tr>
<td>REORG INDEX</td>
<td>displays panels from which you can input options for the IBM REORG INDEX utility</td>
</tr>
<tr>
<td>REPORT</td>
<td>displays panels from which you can input options for the IBM REPORT utility</td>
</tr>
<tr>
<td>REPORT INDEX</td>
<td>displays panels from which you can input options for the IBM REPORT INDEX utility</td>
</tr>
<tr>
<td>RUNSTATS</td>
<td>displays panels from which you can input options for the IBM RUNSTATS utility</td>
</tr>
<tr>
<td>RUNSTATS INDEX</td>
<td>displays panels from which you can input options for the IBM RUNSTATS INDEX utility</td>
</tr>
<tr>
<td>RUNSTATS IX</td>
<td>displays panels from which you can input options for the IBM RUNSTATS IX utility</td>
</tr>
<tr>
<td>SPACE</td>
<td>invokes the BMC DASD MANAGER PLUS product to display space estimation for the object</td>
</tr>
<tr>
<td>STATS</td>
<td>invokes the DASD MANAGER PLUS product to display statistics for the specified object</td>
</tr>
<tr>
<td>STATUS</td>
<td>executes the DB2 DISPLAY UTILITY and BMCUTIL commands, and displays the results in a single list</td>
</tr>
<tr>
<td>STOSPACE</td>
<td>displays panels from which you can input options for the IBM STOSPACE utility</td>
</tr>
<tr>
<td>TERM</td>
<td>executes the DB2 TERM UTILITY command</td>
</tr>
<tr>
<td></td>
<td>The command is valid only from a utility list.</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>displays panels from which you can input options for the IBM UNLOAD utility</td>
</tr>
</tbody>
</table>
Utility list commands

Table 62  Utility commands (part 4 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTIL</td>
<td>displays a panel from which you can select and order one or more utilities to run in a single job for the specified object</td>
</tr>
<tr>
<td>UTILITY profileID</td>
<td>displays panels from which you can input options for the utilities that are included in the specified utility profile</td>
</tr>
</tbody>
</table>

Table 63 lists commands that are valid from the Cmd column in the Utility List panel. CATALOG MANAGER displays the Utility List panel after you type a utility command (see Table 62) on a list.

Table 63  Utility list commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>repeats the command on the previous object</td>
</tr>
<tr>
<td>+ (plus sign)</td>
<td>displays a list of objects that the SS command combined</td>
</tr>
<tr>
<td></td>
<td>Object names that have been combined are no longer displayed on the utility statement list.</td>
</tr>
<tr>
<td>?</td>
<td>displays a list of available commands for the Cmd column</td>
</tr>
<tr>
<td>D</td>
<td>deletes a utility statement from the utility statement list and from the utility job</td>
</tr>
<tr>
<td></td>
<td>The remaining statements are not renumbered.</td>
</tr>
<tr>
<td>ED</td>
<td>displays a panel from which you can edit the parameters for a utility statement and specify option values</td>
</tr>
<tr>
<td>P</td>
<td>displays a panel from which you can select a utility profile for the current utility or delete a utility profile</td>
</tr>
<tr>
<td>RP</td>
<td>resets the utility profile for the current utility</td>
</tr>
<tr>
<td></td>
<td>After you select a utility profile on the Utility List panel, you cannot reset the profile until you exit the panel or issue the RESET command on the Command line.</td>
</tr>
<tr>
<td>SS</td>
<td>combines a utility statement with other unmodified statements for the same utility into a single control statement</td>
</tr>
<tr>
<td></td>
<td>Some utilities, such as IBM UNLOAD, do not allow single statements for multiple objects.</td>
</tr>
</tbody>
</table>

Statistics commands

CATALOG MANAGER can display statistical information from various catalog tables. To execute a command, type the command in the Cmd column next to an appropriate item on a list. Table 64 lists the statistics commands.
List commands

CATALOG MANAGER provides the commands that generate and manipulate lists. In addition to these list commands, you can use the two-letter codes displayed across the top of each list panel to produce other lists. The codes that are available depend on the type of object that the product displays.

Table 65 lists all of the codes and describes the type of list produced.

Table 64  Statistics commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATSTATS</td>
<td>executes the SQL COUNT (*) statement for objects in the DB2 catalog</td>
</tr>
<tr>
<td>COLSTATS</td>
<td>displays statistics from the SYSIBM.SYSCOLSTATS catalog table for the selected table or column</td>
</tr>
<tr>
<td>DEST</td>
<td>displays statistical information and catalog row information for the selected object</td>
</tr>
<tr>
<td>DESTATISTICS</td>
<td></td>
</tr>
<tr>
<td>DISTSTATS</td>
<td>displays statistics from the SYSIBM.SYSCOLDIST or SYSIBM.SYSCOLDISTSTATS catalog table for the selected object</td>
</tr>
<tr>
<td>HISTORY</td>
<td>displays statistics from the SYSIBM history tables for the selected object</td>
</tr>
<tr>
<td>INDEXSTATS</td>
<td>displays statistical information from the SYSIBM.SYSINDEXSTATS catalog table for the selected object</td>
</tr>
<tr>
<td>PDISTSTATS</td>
<td>displays statistics from the SYSIBM.SYSCOLDISTSTATS catalog table for the selected object</td>
</tr>
<tr>
<td>TABSTATS</td>
<td>displays statistics from the SYSIBM.SYSTABSTATS catalog table for the selected object</td>
</tr>
</tbody>
</table>

Table 65  List commands (part 1 of 5)

<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>lists the aliases for the selected object</td>
</tr>
<tr>
<td>APO</td>
<td>lists the audit policies from the SYSIBM.SYSAUDITPOLICIES table</td>
</tr>
<tr>
<td>ARH</td>
<td>lists historical information about autonomic stored procedures in the SYSIBM.SYSAUTORUNS_HIST table</td>
</tr>
<tr>
<td>ATS</td>
<td>lists statistics alerts from autonomic stored procedures in the SYSIBM.SYSAUTOALERTS table</td>
</tr>
<tr>
<td>ATW</td>
<td>lists time windows for running autonomic stored procedures in the SYSIBM.SYSAUTO TIME WINDOWS table</td>
</tr>
<tr>
<td>AU</td>
<td>lists the authorizations</td>
</tr>
<tr>
<td>BP</td>
<td>lists the buffer pool privileges</td>
</tr>
</tbody>
</table>
## List commands (part 2 of 5)

<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR</td>
<td>invokes the options for the CATALOG MANAGER data browsing function. To invoke the IBM DB2 data editor (if it is installed) and browse data from TB, VW, SY, and AL lists, uncomment the BR command in the CATALOG MANAGER commands table.</td>
</tr>
<tr>
<td>CA</td>
<td>lists the column authorizations</td>
</tr>
<tr>
<td>CD</td>
<td>lists the check constraint dependencies</td>
</tr>
<tr>
<td>CI</td>
<td>lists the collection IDs</td>
</tr>
<tr>
<td>CK</td>
<td>lists the check constraints</td>
</tr>
<tr>
<td>CL</td>
<td>lists the column labels</td>
</tr>
<tr>
<td>CO</td>
<td>lists the columns</td>
</tr>
<tr>
<td>CP</td>
<td>lists the constraint dependencies</td>
</tr>
<tr>
<td>CX</td>
<td>lists the trusted contexts</td>
</tr>
<tr>
<td>CXA</td>
<td>lists the authorization IDs for trusted contexts</td>
</tr>
<tr>
<td>CXT</td>
<td>lists the attributes for trusted contexts</td>
</tr>
<tr>
<td>C2</td>
<td>lists the check constraints (for tables created in DB2 Version 7 or later)</td>
</tr>
<tr>
<td>DB</td>
<td>lists the databases</td>
</tr>
<tr>
<td>DM</td>
<td>lists the DBRMs</td>
</tr>
<tr>
<td>DP</td>
<td>lists the dependencies</td>
</tr>
<tr>
<td>DS</td>
<td>lists the data sets. When used with the SYSPROC.ADMIN_DS_LIST stored procedure, the DS command can display data set information on a remote SSID.</td>
</tr>
<tr>
<td>DT</td>
<td>lists the data types</td>
</tr>
<tr>
<td>ED</td>
<td>invokes the options for the CATALOG MANAGER data editing function. To invoke the IBM DB2 data editor (if it is installed) and edit data from TB, VW, SY, and AL lists, uncomment the ED command in the CATALOG MANAGER commands table.</td>
</tr>
<tr>
<td>EN</td>
<td>lists the environments</td>
</tr>
<tr>
<td>FK</td>
<td>lists the foreign keys</td>
</tr>
<tr>
<td>FN</td>
<td>lists the routines (functions)</td>
</tr>
<tr>
<td>FO</td>
<td>lists the routine (function) options</td>
</tr>
<tr>
<td>FP</td>
<td>lists the function parameters</td>
</tr>
<tr>
<td>FS</td>
<td>lists the SQL procedure sources</td>
</tr>
<tr>
<td>IC</td>
<td>lists the image copies</td>
</tr>
<tr>
<td>IL</td>
<td>lists the IP addresses for a given location</td>
</tr>
<tr>
<td>IM</td>
<td>displays a mixed list with each index on one line, and each key column on a following line in key order</td>
</tr>
<tr>
<td>IN</td>
<td>lists the location names and IP addresses of remote systems</td>
</tr>
<tr>
<td>IP</td>
<td>lists the index partitions</td>
</tr>
<tr>
<td>IS</td>
<td>lists the index spaces</td>
</tr>
</tbody>
</table>
**Table 65  List commands (part 3 of 5)**

<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISS</td>
<td>lists the index space statistics from the SYSIBM.SYSINDEXSPACESTATS catalog table, which stores real-time statistics (RTS)</td>
</tr>
<tr>
<td>IX</td>
<td>lists the indexes</td>
</tr>
<tr>
<td>JB</td>
<td>lists the objects from the installed JAR</td>
</tr>
<tr>
<td>JC</td>
<td>lists the Java class information for the installed JAR</td>
</tr>
<tr>
<td>JP</td>
<td>lists the build options that were used to build the installed JAR</td>
</tr>
<tr>
<td>JT</td>
<td>lists the Java paths</td>
</tr>
<tr>
<td>KC</td>
<td>lists the key columns</td>
</tr>
<tr>
<td>KT</td>
<td>displays the contents of the SYSIBM.SYSKEYTARGETS catalog table, which contains key-targets of extended indexes</td>
</tr>
<tr>
<td>KTD</td>
<td>displays the contents of the SYSIBM.SYSKEYTGTDIST catalog table, which contains distribution information for key-targets of extended index keys</td>
</tr>
<tr>
<td>KTDH</td>
<td>displays the contents of the SYSIBM.SYSKEYTGTDIST_HIST catalog table, which contains history information for key-targets of extended index keys</td>
</tr>
<tr>
<td>KTDS</td>
<td>displays the contents of the SYSIBM.SYSKEYTGTDISTSTATS catalog table, which contains the key-targets of data-partitioned secondary indexes</td>
</tr>
<tr>
<td>KTH</td>
<td>displays the contents of the SYSIBM.SYSKEYTARGETS_HIST catalog table, which contains history information for key-targets of extended indexes</td>
</tr>
<tr>
<td>KTS</td>
<td>displays the contents of the SYSIBM.SYSKEYTARGETSTATS catalog table, which contains partition statistics for key-targets of extended index keys</td>
</tr>
<tr>
<td>KU</td>
<td>lists the constraint key columns</td>
</tr>
<tr>
<td>LI MX objectType qualifier</td>
<td>when issued from the Command line, displays a secondary list that shows multiple object types associated with certain source object types</td>
</tr>
<tr>
<td></td>
<td>The object name must be fully qualified.</td>
</tr>
<tr>
<td>LIST</td>
<td>when entered in the Cmd column of a list line, produces a sublist qualified by the object name on the current list</td>
</tr>
<tr>
<td></td>
<td>Using LIST from the Command line discards all current lists and creates a new level-one list.</td>
</tr>
<tr>
<td></td>
<td>LIST is also an ISPF command. To use the command on the Command line, you must either abbreviate the command (enter LI or LIS) or precede it with the command recognition character.</td>
</tr>
<tr>
<td>LK</td>
<td>lists the limit keys</td>
</tr>
<tr>
<td>LL</td>
<td>lists the logical unit (LU) names for a location from the SYSIBM.LULIST table</td>
</tr>
<tr>
<td>LM</td>
<td>lists the limits for LUNAMEs and MODENAMEs from the SYSIBM.LUMODES table</td>
</tr>
<tr>
<td>LO</td>
<td>lists the accessible remote locations from the SYSIBM.LOCATIONS table</td>
</tr>
<tr>
<td>LS</td>
<td>lists the mode names for SQL requests from the SYSIBM.LUMODESELECT table</td>
</tr>
<tr>
<td>LU</td>
<td>lists the remote locations that communicate with DB2 from the SYSIBM.LUNAMES table</td>
</tr>
<tr>
<td>MK</td>
<td>lists the column masks from the SYSIBM.SYSCONTROLS table</td>
</tr>
<tr>
<td>MQT</td>
<td>lists the materialized query tables</td>
</tr>
</tbody>
</table>
### Table 65  List commands (part 4 of 5)

<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX</td>
<td>displays a mixed list showing objects that are dependent on the selected object</td>
</tr>
<tr>
<td>NP</td>
<td>lists the native SQL stored procedures</td>
</tr>
<tr>
<td>OB</td>
<td>lists the online schema changes</td>
</tr>
<tr>
<td>OS</td>
<td>displays the contents of the SYSIBM.SYSLOBSTATS catalog table, which contains the statistics from LOB table spaces</td>
</tr>
<tr>
<td>PA</td>
<td>lists the plan authorizations</td>
</tr>
<tr>
<td>PDD</td>
<td>lists information about the objects that have pending changes to data definitions from the SYSIBM.SYSPENDINGDDL table</td>
</tr>
<tr>
<td>PG</td>
<td>lists the packages</td>
</tr>
<tr>
<td></td>
<td>The PG command can be issued on a sequence object in a Sequences List. The dependency of the package on the sequence object must be specified in the SYSIBM.SYSPACKDEP catalog table.</td>
</tr>
<tr>
<td>PGC</td>
<td>lists the package copies from the SYSIBM.SYSPACKCOPY table</td>
</tr>
<tr>
<td>PI</td>
<td>lists the packages that a plan can use</td>
</tr>
<tr>
<td>PK</td>
<td>lists the primary keys</td>
</tr>
<tr>
<td>PL</td>
<td>lists the plans</td>
</tr>
<tr>
<td>PM</td>
<td>lists the row permissions from the SYSIBM.SYSCONTROLS table</td>
</tr>
<tr>
<td>PR</td>
<td>lists the procedures</td>
</tr>
<tr>
<td>PT</td>
<td>lists the partitions</td>
</tr>
<tr>
<td>QRO</td>
<td>lists the optimization parameters for queries from the SYSIBM.SYSQUERYOPTS table</td>
</tr>
<tr>
<td>QRP</td>
<td>lists the plan hint information for queries from the SYSIBM.SYSQUERYPLAN table</td>
</tr>
<tr>
<td>QRY</td>
<td>lists the queries from the SYSIBM.SYSQUERY table</td>
</tr>
<tr>
<td>RD</td>
<td>lists the object role dependencies</td>
</tr>
<tr>
<td>RE</td>
<td>lists the referential constraints from the SYSIBM.SYRELSTM table</td>
</tr>
<tr>
<td>RI</td>
<td>displays the referential integrity list, which shows all tables and relations contained in the selected object</td>
</tr>
<tr>
<td></td>
<td>The RI list also displays an asterisk (*) on the rows in which an object is included in the selected object.</td>
</tr>
<tr>
<td>RO</td>
<td>lists the object roles</td>
</tr>
<tr>
<td>SC</td>
<td>lists the schemas</td>
</tr>
<tr>
<td>SE</td>
<td>lists the identity columns in sequences from the SYSIBM.SYSEQUENCES table</td>
</tr>
<tr>
<td>SG</td>
<td>lists the storage group objects</td>
</tr>
<tr>
<td>ST</td>
<td>lists the strings</td>
</tr>
<tr>
<td>SU</td>
<td>lists the system privileges for the AUTHID pattern that you entered</td>
</tr>
<tr>
<td>SY</td>
<td>lists the synonyms</td>
</tr>
<tr>
<td>TB</td>
<td>lists the tables</td>
</tr>
<tr>
<td>TBP</td>
<td>lists the table profiles from the SYSIBM.SYSTABLES_PROFILES catalog table</td>
</tr>
<tr>
<td>TC</td>
<td>lists the column authorizations</td>
</tr>
</tbody>
</table>
User commands

CATALOG MANAGER can execute user commands. To execute a command, type the command in the Cmd column next to an appropriate item on a list. Table 66 lists the user commands.

Table 65  List commands (part 5 of 5)

<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TM</td>
<td>displays a mixed list of table space sets</td>
</tr>
<tr>
<td>TP</td>
<td>lists the table space partitions</td>
</tr>
<tr>
<td>TR</td>
<td>lists the triggers</td>
</tr>
<tr>
<td>TS</td>
<td>lists the table spaces</td>
</tr>
<tr>
<td>TSS</td>
<td>lists the table space statistics from the SYSIBM.SYSTABLESPACESTATS catalog table, which stores real-time statistics (RTS)</td>
</tr>
<tr>
<td>TT</td>
<td>lists the table space sets</td>
</tr>
<tr>
<td>UA</td>
<td>lists the user authorizations</td>
</tr>
<tr>
<td>UN</td>
<td>lists the user names</td>
</tr>
<tr>
<td>US</td>
<td>lists the users (authorization IDs)</td>
</tr>
<tr>
<td>VL</td>
<td>lists the volumes</td>
</tr>
<tr>
<td>VW</td>
<td>lists the views</td>
</tr>
<tr>
<td>XR</td>
<td>lists the XML relationships</td>
</tr>
<tr>
<td>XS</td>
<td>lists the XML strings</td>
</tr>
<tr>
<td>XT</td>
<td>lists the auxiliary and base tables</td>
</tr>
</tbody>
</table>

Table 66  User commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCAT</td>
<td>invokes a CLIST that executes a VSAM LISTCAT command for the selected data set</td>
</tr>
<tr>
<td>TYPES</td>
<td>displays the valid abbreviations for all of the object types</td>
</tr>
</tbody>
</table>
Data browsing and editing commands

Most ISPF commands are valid for the data browsing and data editing features. Specific CATALOG MANAGER commands also facilitate these functions.

Command-line commands

Table 67 lists commands that are valid from the Command line of the Browse DB2 Table and Edit DB2 Table panels. With the scrolling commands, you can press a scroll key instead of the Enter key.

Table 67 Command-line commands for browsing or editing data (part 1 of 4)

<table>
<thead>
<tr>
<th>Command (short form)</th>
<th>Function E=Edit, B=Browse</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CANCEL</td>
<td>E,B</td>
<td>ends the edit or browse session</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In data editing mode, changes are not saved.</td>
</tr>
<tr>
<td>CHANGE</td>
<td>E</td>
<td>locates and modifies a character string that is specified in the command</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The syntax of the command is CHANGE 'targetString' 'replacementString'.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The :columnIdentifier parameter can also be used with the CHANGE command to restrict the search to a single column. Valid :columnIdentifier values are :columnName, :columnNumber, and :columnLabel.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The :columnNumber identifier references the column number as defined in the table structure, not as displayed on the panel.</td>
</tr>
<tr>
<td>COPY</td>
<td>E</td>
<td>inserts rows from a source table into the displayed target table</td>
</tr>
<tr>
<td>DOWN</td>
<td>E, B</td>
<td>in column view, scrolls forward through rows; in row view, scrolls forward through columns</td>
</tr>
<tr>
<td>EDIT</td>
<td>E, B</td>
<td>when the cursor is in a data field, opens an ISPF edit session in which you can change the data in the field</td>
</tr>
<tr>
<td>END (F3)</td>
<td>E, B</td>
<td>ends the edit or browse session</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In data editing mode, saves your changes and ends the edit session normally</td>
</tr>
</tbody>
</table>
FIND (F) E, B locates a character string that you specify in the command. Usually, you enter the FIND command, followed by a space and then the character string. To locate a character string that includes blank characters, you must delimit the character string.

The FIND command searches for the specified string, starting at the top of the display or at the cursor position. If the character string is located, the result of a FIND command places the cursor at the start of the character string. If the character string is not located, either the "Bottom of Data reached" or "Top of Data reached" message is displayed.

The FIND command can search only in columns that can be displayed. Large character and graphic columns might not be displayed in their entirety, therefore a FIND command might not return occurrences of a character string embedded in such column types.

The following parameters can be used with the FIND command. The syntax of the command is FIND characterString PARAMETER.

- NEXT is the default parameter. Using the NEXT parameter returns the same result as using the FIND command with only the character string.
- PREV locates the first occurrence of the character string preceding the cursor location or the first line of the displayed data.
- FIRST searches the data for the first occurrence of the character string.
- LAST searches the data for the last occurrence of the character string.
- ALL searches the data for each occurrence of the character string, beginning at the first line and continuing to the last line.
- :columnIdentifier restricts the search to a single column. Valid column identifiers are :columnName, :columnNumber, and :columnLabel.

Note: The :columnNumber identifier references the column number as defined in the table structure, not as displayed on the panel.

The FIND command is valid only from the Command line.

FREEZE E, B anchors table columns while you edit or browse data.

The columns remain visible when you scroll left and right.

HOME E, B displays the left-most column and first row of the table, and positions the cursor at the Command line.
### Command-line commands for browsing or editing data (part 3 of 4)

<table>
<thead>
<tr>
<th>Command (short form)</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| IM, IX, KC           | E, B     | displays the following information about a table or view:  
|                      |          | - qualified index name  
|                      |          | - cluster information  
|                      |          | - type of key  
|                      |          | - type of index  
|                      |          | - column name in the index  
|                      |          | - position of the column in the key  
|                      |          | - order of the column in the key  
| LEFT nnn LEFT M (F10) | E, B     | in column view, scrolls backward through columns; in row view, scrolls backward through rows  
|                      |          | Type a number nnn after the LEFT command to scroll nnn DB2 columns to the left, or type M(AX) after the LEFT command to scroll to the left-most column.  
|                      |          | To use PF keys, type M or the number nnn, and then press the function key.  
| MORE                 | E, B     | displays additional rows when the address space is insufficient to display all of the selected rows  
| OPT                  | E, B     | displays the Edit and Browse Options panel  
| RCHANGE              | E        | locates and modifies the next occurrence of the character string that is specified in the CHANGE command  
| RESET, RES           | E, B     | clears pending line commands and restores an original numeric value after an incorrect update  
|                      |          | Some CATALOG MANAGER commands, such as COUNT, produce results that are displayed on the list lines where the command was issued.  
| RFIND (F5)           | E, B     | locates the next occurrence of the character string that is specified in the FIND command  
|                      |          | You must enter a FIND command before you can enter the RFIND command.  
| RIGHT nnn RIGHT M (F11) | E, B     | in column view, scrolls forward through columns; in row view, scrolls forward through rows  
|                      |          | Type a number nnn after the RIGHT command to scroll nnn DB2 columns to the right, or type M(AX) after the RIGHT command to scroll to the right-most column.  
|                      |          | To use PF keys, type M or the number nnn, and then press the function key.  
| ROWVIEW, ROW         | E, B     | switches the display from column view to row view  
| SAVE                 | E        | saves your changes without ending the edit session  

---

Table 67 Command-line commands for browsing or editing data (part 3 of 4)

---

472 CATALOG MANAGER for DB2 User Guide
Table 67  Command-line commands for browsing or editing data (part 4 of 4)

<table>
<thead>
<tr>
<th>Command (short form)</th>
<th>Function (E=Edit, B=Browse)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SORT</td>
<td>E, B</td>
<td>displays a panel on which you can select nine column names and identify whether to sort by ascending or descending order. You can also issue the SORT command followed by a column identifier and sort order indicator to perform the SORT on the editing panel, for example, SORT DATE ASC.</td>
</tr>
<tr>
<td>THAW</td>
<td>E, B</td>
<td>reverses the action of the FREEZE command.</td>
</tr>
<tr>
<td>UNDO</td>
<td>E</td>
<td>in row view, reverses the changes made during an edit session.</td>
</tr>
<tr>
<td>UP</td>
<td>E, B</td>
<td>in column view, scrolls backward through rows; in row view, scrolls backward through columns.</td>
</tr>
<tr>
<td>ZOOM (F4)</td>
<td>E, B</td>
<td>opens an ISPF session in which you can view the data in the field that is too large to display in CATALOG MANAGER. In data editing mode, you can also modify the data.</td>
</tr>
</tbody>
</table>

Line commands

Table 68 lists line commands that are valid in the Browse DB2 Table and Edit DB2 Table panels.

---

**NOTE**

Line commands are valid only in Column view mode.

Table 68  Line commands for browsing or editing data (part 1 of 2)

<table>
<thead>
<tr>
<th>Command (short form)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I^a</td>
<td>inserts one or more blank lines below the line on which the command is entered.</td>
</tr>
<tr>
<td>D^a-b</td>
<td>deletes one or more lines.</td>
</tr>
<tr>
<td>R^a-b</td>
<td>inserts a line with the same values as the line on which the command is entered.</td>
</tr>
<tr>
<td>C^a-b</td>
<td>identifies the source line of a copy operation.</td>
</tr>
<tr>
<td>M^a-b</td>
<td>identifies the source line of a move operation.</td>
</tr>
<tr>
<td>A</td>
<td>indicates the line after which the copied or moved data will be inserted.</td>
</tr>
<tr>
<td>B</td>
<td>indicates the line before which the copied or moved data will be inserted.</td>
</tr>
</tbody>
</table>
Table 68  Line commands for browsing or editing data (part 2 of 2)

<table>
<thead>
<tr>
<th>Command (short form)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U       ^        ^</td>
<td>reverses the changes that you have made since the most recent SAVE</td>
</tr>
</tbody>
</table>
| Z. V   !            | displays the data in the row on which the command is entered vertically rather than horizontally, with one column per line  
|                     | This vertical row format is called row view. By using row view, you can display columns that are too large to display in column view.  
|                     | To return the display to column view, press END. |

\^ You can follow these commands with a numeric value to apply the command multiple times.

\^ These commands can be used with block identifiers. For example, you can enter DD on two lines to mark those lines and the lines between them for deletion.
Glossary

A

ACT
The product code that BMC uses to identify the CATALOG MANAGER for DB2 product.

Administrative Products for DB2
A collection of products from BMC that includes ALTER, CATALOG MANAGER, CHANGE MANAGER, and DASD MANAGER PLUS. These integrated products are designed to help database administrators, system programmers, and application developers automate the tasks associated with the implementation and administration of a DB2 Universal Database for z/OS system.

When all Administrative products for DB2 are installed properly, they can access some of each other’s functionality. In addition, these products can access the IBM and BMC utilities.

AEXIN
The ddname of the input stream that the Execution component uses.

AEXPRINT
The ddname of diagnostic output of the Execution component. This diagnostic output data set contains all output from the Execution process, including DB2 for z/OS utility messages, BMC utility messages, dynamic SQL messages, IDCAMS messages, and any other messages that are generated by the actions of the worklist. AEXPRINT is frequently referred to as the worklist execution log.

AJX variables
A group of variables (JOB, STEP, and DD) that are used during Job Control Language (JCL) generation. JOB global variables are set once per JCL creation session. STEP global variables are set at the beginning of a JCL creation session and are updated when a new JOBSTEP is detected. Data Definitions (DD) variables are local variables that are set for each creation of a JCL DD entry. Descriptions of these variables are provided in HLQ.SLIB($AJXDOCV).

AJXPOFVL
The ddname of the product options file (POF) validation report that the Batch Execution JCL Generation component uses.

AJXPRINT
The ddname of the diagnostic output of the Batch Execution JCL Generation component.
alloc unit
   The allocation unit that is used for space estimation calculations, volume placement, and primary and secondary quantities. Possible values are K (kilobytes), T (tracks), or C (cylinders). The default value comes from the default option modules.

ALTER for DB2
   A BMC product that provides advanced database administration and manipulation within a single DB2 subsystem. ALTER streamlines the process of changing and migrating database objects, handles analysis for both changes and migrations, and automatically generates SQL, DB2 commands, and utilities.

B

BMCCOPY
   The BMC COPY PLUS utility that is used to create an image copy. The short form of the command is BMCI.

BMCLOAD
   The BMC LOADPLUS utility that is used to load DB2 tables. The short form of the command is BMCL.

BMCREORG
   The BMC REORG PLUS utility that is used to perform a reorganization. The short form of the command is BMCR.

BMCSTATS
   A command that invokes the statistics collecting function of the DASD MANAGER PLUS product. The short form of the command is BMCS. BMCSTATS is similar to the IBM RUNSTATS utility. It provides the same statistics plus many additional statistics.

BMCUNLOAD
   The BMC UNLOAD PLUS utility that is used to unload data from a full image copy of one or more tables in a table space. The short form of the command is BMCD.

C

CANCEL (CAN)
   An ISPF command that returns you to the previous panel without saving any changes made to the current panel.

catalog indirection
   An optional method of implementing the BMC Administrative products for DB2 that allows them to access the DB2 catalog indirectly when making information queries. Catalog indirection is accomplished by using synonyms that point either to a copy of the DB2 catalog or to user-created views of the catalog. Benefits include reducing catalog contention and providing an additional level of security for sensitive catalog data.
CATALOG MANAGER for DB2
A BMC product that automates the day-to-day tasks associated with administering DB2. This product provides an interactive, intuitive, and easy-to-use interface for submitting DB2 commands and retrieving catalog information using qualified lists, wildcard searches, and dependency lists. CATALOG MANAGER provides the ability to create and drop DB2 objects and re-create dropped structures and data.

CHANGE MANAGER for DB2
A BMC product that enables database administrators, system administrators, and developers to manage user applications and individual database objects globally. As a robust extension of the ALTER product, CHANGE MANAGER automates data structure changes across multiple DB2 subsystems and between DB2 and data modeling tools by providing a way to implement, migrate, synchronize, and back out data structure changes while preserving structure modifications that might be unique to a specific subsystem.

CLIST
Command list.

command
A token that you can enter at the command prompt on a panel.

See also action code.

D

DASD MANAGER PLUS for DB2
A BMC product that automates utility generation, gathers comprehensive statistics, monitors changes in the database, and enables you to perform maintenance based on the condition of the data instead of a rigid schedule.

Data Control Language (DCL)
A category of SQL statements that control data security.

Data Definition Language (DDL)
A category of SQL statements that create, modify, or delete database objects.

data definition name (ddname)
The name of a data definition (DD) statement in job control language (JCL) that corresponds to a data control block that contains the same name.

data set sizing
The process of determining data set allocations, especially as used when generating JCL. Data set sizing is distinct from space estimation.

database administrator (DBA)
An individual who is responsible for the design, development, operation, security, maintenance, and use of databases.
**DB2 command**
An instruction to the DB2 subsystem. Some example command processes enable you to start or stop DB2, display information on current users, start or stop databases, and display information about databases. DB2 commands always begin with a hyphen (–).

**DD statement**
Data Definition statement.

**DDL**
See Data Definition Language.

**ddname**
See data definition name.

**default options (DOPTS)**
See installation options.

**default value**
A predetermined value, attribute, or option that is assumed when no other is explicitly specified.

**dependencies**
The name or values of objects which another object uses as part of its definition or as a hierarchical subordinate.

**DOPTS**
See installation options.

**DSN1COPY**
A utility command that invokes the IBM DSN1COPY utility. The short form of the command is DSN1.

**E**

**END**
An ISPF command, similar to Enter, that validates and processes the information on a panel and returns you to the previous panel, but does not execute commands. This command is typically programmed on the keyboard as a function key, such as PF3.

**Enter key**
The key that executes any commands that have been specified. For a sequence of panels, the Enter key displays the next panel.

**exit routine**
A program (BMC, IBM, or user-written) that receives control from DB2 to perform specific functions. Exit routines run as extensions of DB2 (for example, authorization checking).
**G**

**generation data group (GDG)**
A finite number of data sets that are kept in chronological order. Each data set is a generation data set.

**H**

**hash value**
A number that appears at the end of commands in worklists. The hash value is generated based on the contents of the command line and allows the products to determine if the line has been manually modified since the file was generated.

**HLQ**
High-level qualifier of a data set.

**I**

**image copy**
An exact reproduction of all or part of the data in a table space. IBM provides utility programs to make full image copies (copy the entire table space) or incremental image copies (copy only the pages that have been modified since the last image copy). The BMC COPY PLUS utility can perform the same function.

**incremental DDL**
The DDL that changes the data structures that exist in the DB2 catalog, by using either an alter strategy or a drop-then-rebuild strategy.

**installation options**
An assembler module that contains keywords whose global values determine the operating environment for a BMC product.

**J**

**JCL DSN**
The name of the data set that contains job control language (JCL). The data set must exist and can be partitioned or sequential. You must specify a member name for partitioned data sets. You can use symbolic variables.

**JCL Generation (JCLGEN)**
A component of some BMC products that constructs a job control language (JCL) file for executing processes in batch mode.

**JCL variable display**
The resulting output of a user option that includes debugging comments within any generated JCL. All AJX-prefixed variables are displayed as //* comments in the JCL to assist in diagnosing JCL Generation problems.
M

menu
A list of action options for the ISPF interface. You select an action by typing its corresponding number in the option input field and pressing Enter. A menu panel might contain other fields that you can use to qualify the action.

O

option
A named value that is used to control one or more components. Global options are defined in the installation options module. The user can override the installation options by specifying the user options or by specifying keywords in the component’s AEXIN or AJXIN input stream.

R

recovery
The process of restoring a set of data structure definitions to their state at a particular point in time.

S

selection list
A list of related items from which you can select one for further action. The actions (line commands) that you can specify in the Act field are typically displayed across the top of the panel.

SQL ID
The authorization ID that is used as the implicit qualifier of table, view, synonym, and index names in dynamic SQL statements. The SQL ID, along with the other authorization IDs of a process, is used for authorization checking of dynamic SQL statements.

SSID
A DB2 subsystem identifier.

sequence number
A six-digit, zero-filled number that identifies a statement in a worklist. The sequence number appears in columns 7 through 12 of the first line of each worklist command.

Structured Query Language (SQL)
An ANSI-standard language for database definition, manipulation, security, and query.

symbolic variable
A user interface variable that has its value set interactively for the current user and session at the time of job control language (JCL) generation. The BMC JCL Generation component (JCLGEN) uses symbolic variables to perform ISPF file tailoring services. A symbolic variable should be
preceded with an ampersand (&). In the default options modules, a symbolic variable should be preceded with two ampersands (&&).

Note: Symbolic variables are not global job variables, which have values that are set for all users and all sessions.

See also JCL Generation.

V

variable
See symbolic variable.

W

wildcard
A symbol that you can use to represent a value in SQL statements, filters, and name patterns. Valid wildcards for SQL statements and filters include the following symbols:

- % and * represent any string of one or more characters
- ? and _ represent a single character

worklist
A data set that contains commands for implementing a data structure change or migration.
Index

Symbols

! line command 474
$ACTCMD macro 145
$ACTEXC macro 152, 153
$ACTLIST macro 152
$ACTULOG macro 152
$ACTVARS macro 152
$AJXDOC data set 362
&ACTCOLID ISPF variable 156
&ACTSRVR ISPF variable 156
&AJXLLQ AJXPOFIN keyword 415
&AJXRHLQ SLIB variable 431
&AJXSYM SLIB variable 433
&AJXULLQ SLIB variable 436
&CLIST parameter 145
&CMD parameter 146
&DB2MAX parameter 146
&DB2MIN parameter 146
&HELP parameter 146
&LOAD parameter 146
&LSTO parameter 146
&NLIST parameter 146
&NOSERVER parameter 147
&NUM parameter 147
&OBJECTS parameter 147
&PARSE parameter 147
&PLAN parameter 147
&SSID ISPF variable 156
&VCAT ISPF variable 156
&WFEK parameter 148
**PREFIX** TEMPLATE descriptor variable
  PREFIX symbolic variable 448
  SYSUID symbolic variable 450
  UID symbolic variable 451
  USERID symbolic variable 451
  ZPREFIX symbolic variable 453
  ZSYSID symbolic variable 454
  ZUSER symbolic variable 454
*DROP marker 278
*PERSIST option 82
+ command 464
+1 OUTPUT descriptor variable 443
+1 TEMPLATE descriptor variable 443
=X command 46
? command 464

Numerics

10-byte object type address (TYPE) 154
16-byte function address (FUNC) 154
18-byte object name address (NAME) 154
2MEGSQL AEXIN keyword 383
2MEGSQL AJXPOFIN keyword 390
2SQL command 280
2WL command 455
8-byte object name qualifier (QUAL) 154

A

A line command 473
above-the-bar storage 71
ACM AEXIN keyword 383
ACM_AMS POF keyword 390
ACM_BASDIAG POF keyword 390
ACM_BRPTDIAG POF keyword 390
ACM_BRPTDSN POF keyword 390
ACM_CDLDSN POF keyword 391
ACM_CDLPS POF keyword 391
ACM_CDLSS POF keyword 391
ACM_CDLU POF keyword 391
ACM_CMPDIAG POF keyword 391
ACM_CPLCDLO POF keyword 391
ACM_CPLDIAG POF keyword 391
ACM_CPLWDSN POF keyword 392
ACM_CPLWDSNO POF keyword 392
ACM_DBRM1 POF keyword 392
ACM_DBRM2 POF keyword 392
ACM_DBRM3 POF keyword 392
ACM_GLID POF keyword 392
ACM_IMPDIAG POF keyword 393
ACM_JDSN POF keyword 393
ACM_JDSNB POF keyword 393
ACM_JDSNBG POF keyword 393
ACM_JDSNBR POF keyword 393
ACM_JDSNC POF keyword 394
ACM_JDSNCPL POF keyword 394
ACM_JDSNCPOLO POF keyword 394
ACM_JDSNE POF keyword 394
ACM_JDSNI POF keyword 394
ACM_PARALLEL_MAXINIT AJXPOFIN keyword 395
ACM_PARALLEL_MININIT AJXPOFIN keyword 395
ACM_PARALLEL_WORKLIST AJXPOFIN keyword 395
ACM_PARALLEL_XIMGRP AJXPOFIN keyword 395
ACM_PARALLEL_XIMPROC AJXPOFIN keyword 395
ACM_PARALLEL_XIMSTRT AJXPOFIN keyword 395
ACM_PARALLEL_XIMTRCE AJXPOFIN keyword 396
ACM_PIC POF keyword 396
ACM_SDSN POF keyword 396
ACM_WLORDER POF keyword 397
ACM_WLORDERMSG POF keyword 397
ACT product code 69
ACTCOMND member 143, 152
ACTIVATE command 459
ACTPSS CLIST 374
ACTUSER member 152
ACTvrDB Bind/Rebind plan 69
ACTvrDE Data Editor plan 69
ACTvrDG Generate SQL plan 69
ACTvrDH Utility Status Display plan 69
ACTvrDK Command Generation and Execution plan 69
ACTvrDL Log Table maintenance plan 69
ACTvrDM Display DB2 Catalog plan 69
ACTvrDS Search plan 70
ACTvrDU Grant Authorities plan 70
ADMIN_COMMAND_DB2 stored procedure 164, 169, 170
ADMIN_DS_LIST stored procedure 164
Administrative Assistant for DB2 30
AEXIN input stream, user options 72
AEXIN keywords, list of 383–389
AJXAREX SLIB variable 398
AJXARMCL SLIB variable 398
AJXARPRF SLIB variable 398
AJXARRP SLIB variable 398
AJXARSCL SLIB variable 398
AJXARTAD SLIB variable 398
AJXARTAM SLIB variable 398
AJXARTAS SLIB variable 398
AJXARTHX SLIB variable 399
AJXBF SLIB variable 399
AJXBLALU SLIB variable 401
AJXBLDCL SLIB variable 400
AJXBLEX SLIB variable 400
AJXBLMCL SLIB variable 400
AJXBLPRF SLIB variable 400
AJXBLRP SLIB variable 400
AJXBLSCL SLIB variable 400
AJXBLTAD SLIB variable 400
AJXBLTAM SLIB variable 400
AJXBLTAS SLIB variable 400
AJXBLTHX SLIB variable 400
AJXBMCCP SLIB variable 442
AJXBRPS SLIB variable 400
AJXBRSS SLIB variable 400
AJXBRUNI SLIB variable 401
AJXC1ALU SLIB variable 418
AJXC1DCL SLIB variable 417
AJXC1EX SLIB variable 417
AJXC1MCL SLIB variable 417
AJXC1PRF SLIB variable 417
AJXC1RP SLIB variable 417
AJXC1SCL SLIB variable 418
AJXC1SF SLIB variable 418
AJXC1TAD SLIB variable 417
AJXC1TAM SLIB variable 417
AJXC1TAS SLIB variable 418
AJXC1THX SLIB variable 418
AJXC2ALU SLIB variable 419
AJXC2DCL SLIB variable 418
AJXC2EX SLIB variable 418
AJXC2MCL SLIB variable 418
AJXC2PRF SLIB variable 418
AJXC2PS SLIB variable 418
AJXC2RP SLIB variable 419
AJXC2SF SLIB variable 419
AJXC2SS SLIB variable 419
AJXC2SK SLIB variable 419
AJXC2TAD SLIB variable 418
AJXC2TAM SLIB variable 418
AJXC2TAS SLIB variable 419
AJXC2THX SLIB variable 419
AJXC2UNI SLIB variable 419
AJXCLDCL SLIB variable 404
AJXCLEX SLIB variable 404
AJXCLMCL SLIB variable 404
AJXCLNRC SLIB variable 404

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Index

AJXCLPRF SLIB variable 404
AJXCLPS SLIB variable 404
AJXCLSRD SLIB variable 404
AJXCLSLC SLIB variable 404
AJXCLSS SLIB variable 404
AJXCLUDI SLIB variable 404
AJXCR SLIB variable
  CR symbolic variable 442
  IXCR symbolic variable 444
AJXDB SLIB variable 442
AJXDBV2 SLIB variable 442
AJXDBV3 SLIB variable 443
AJXDBGSC SLIB variable 433
AJXDBGSS SLIB variable 408
AJXDBGVC SLIB variable 408
AJXDCALU SLIB variable 407
AJXDCDCL SLIB variable 406
AJXDCETX SLIB variable 406
AJXDCMCL SLIB variable 406
AJXDCPRF SLIB variable 406
AJXDCPS SLIB variable 406
AJXDCSCL SLIB variable 407
AJXDCSS SLIB variable 407
AJXDCSTAD SLIB variable 406
AJXDCSTM SLIB variable 406
AJXDCSTAS SLIB variable 407
AJXDCSTHX SLIB variable 407
AJXDCUNI SLIB variable 407
AJXDDN SLIB variable 444
AJXDDOPT SLIB variable 443
AJXDDSQC SLIB variable
  DDSEQ symbolic variable 443
  SEQ symbolic variable 449
  SQ symbolic variable 449
AJXDGDL SLIB variable 405
AJXDGKLL SLIB variable 405
AJXDRNIC POF keyword 408
AJXDNS4 SLIB variable 409, 442
AJXDWU4I SLIB variable 405
AJXERALU SLIB variable 410
AJXERDCL SLIB variable 409
AJXEREX SLIB variable 409
AJXERMCL SLIB variable 409
AJXERPFR SLIB variable 409
AJXERPS SLIB variable 409
AJXERRF SLIB variable 409
AJXERSCL SLIB variable 410
AJXERSSL SLIB variable 410
AJXERTAD SLIB variable 409
AJXERTAM SLIB variable 409
AJXERTNML SLIB variable 410
AJXERUNI SLIB variable 410
AJXESTI SLIB variable 405
AJXEXPDT SLIB variable 434
AJXFCDL SLIB variable 443
AJXFDC SLIB variable 410
AJXFEX SLIB variable 410
AJXFJCHSLIB variable 445
AJXFJCLSLIB variable 411
AJXFLEX SLIB variable 411
AJXFLMCL SLIB variable 411
AJXFLPRF SLIB variable 411
AJXFLPS SLIB variable 412
AJXFLRP SLIB variable 412
AJXFLSLC SLIB variable 412
AJXFLSS SLIB variable 412
AJXFLUNI SLIB variable 412
AJXFMC SLIB variable 410
AJXFPR SLIB variable 410
AJXFPS SLIB variable 411
AJXFPF SLIB variable 411
AJXFRP SLIB variable 411
AJXFSCL SLIB variable 411
AJXFSF SLIB variable 411
AJXFSSS SLIB variable 411
AJXFTSOC SLIB variable 435
AJXGDGB SLIB variable 405
AJXGDGL2 SLIB variable 406
AJXGDGMD SLIB variable 412
AJXGDGPC SLIB variable 443
AJXHF SLIB variable 412
AJXHM SLIB variable 443
AJXHMS SLIB variable
  HMS symbolic variable 443
  HO symbolic variable 444
  HOUR symbolic variable 444
  JHMS symbolic variable 444
  MI symbolic variable 447
  MINUTE symbolic variable 447
  SC symbolic variable 449
  SEC symbolic variable 449
  SECOND symbolic variable 449
  TI symbolic variable 450
  TIME symbolic variable 450
  TIME4 symbolic variable 450
AJXIC SLIB variable 444
AJXIKJ SLIB variable 435
AJXIN input stream 361
AJXIX SLIB variable 444
AJXIXNOD SLIB variable 444
AJXIXSPC SLIB variable 444
AJXJACKL SLIB variable 401
AJXJACML SLIB variable 403
AJXJACPL SLIB variable 401
AJXJACTL SLIB variable 403
AJXJADD1 SLIB variable 397
AJXJADD2 SLIB variable 398
AJXJADUL SLIB variable 403
AJXJAECL SLIB variable 410
AJXJAFRL SLIB variable 402
AJXJAID SLIB variable 442
AJXJAMUL SLIB variable 401
AJXJARUL SLIB variable 402
AJXASUL SLIB variable 405
AJXJB1 SLIB variable 413
AJXJB2 SLIB variable 413
AJXJB3 SLIB variable 413
AJXJB4 SLIB variable 413
AJXJB5 SLIB variable 413
AJXBLIB SLIB variable 412
AJXDDDN SLIB variable 443
AJJINC SLIB variable 432
AJXJULD SLIB variable 445
AJXJBNM SLIB variable 445
AJXJBT SLIB variable 445
AJXJPCOD SLIB variable 445
AJXJQID SLIB variable
  JQID symbolic variable 446
  WKID symbolic variable 452
  WORKID symbolic variable 452
AJXJSQLL SLIB variable 431
AJXJSSID SLIB variable
  ATTACH symbolic variable 442
  GRPNM symbolic variable 443
  SS symbolic variable 449
AJXJYMD SLIB variable 446
AJXKDOPT SLIB variable 401
AJXLDEFN SLIB variable 446
AJXLDJOIN SLIB variable 402
AJXLSUT SLIB variable 415
AJXLLC SLIB variable 413
AJXLLC2 SLIB variable 413
AJXLLC3 SLIB variable 413
AJXLLC4 SLIB variable 413
AJXLLC5 SLIB variable 413
AJXLLL SLIB variable 413
AJXLLL2 SLIB variable 413
AJXLLL3 SLIB variable 413
AJXLLL4 SLIB variable 413
AJXLLL5 SLIB variable 413
AJXLLM SLIB variable
  LOCREM symbolic variable 446
  LR symbolic variable 446
  TYPE symbolic variable 451
AJXMAXPR SLIB variable 416
AJXMAXQT SLIB variable 416
AJXMAXSS SLIB variable 417
AJXMEMBER SLIB variable 447
AJXMEMBR SLIB variable 447
AJXMLMT SLIB variable 417
AJXMPALU SLIB variable 416
AJXMPDCL SLIB variable 415
AJXMPLEX SLIB variable 415
AJXMPMCL SLIB variable 415
AJXMPPRF SLIB variable 415
AJXMPPS SLIB variable 416
AJXMPRP SLIB variable 416
AJXMPSSLIB variable 416
AJXMPSS SLIB variable 416
AJXMPPSLIB variable 416
AJXMPAD SLIB variable 415
AJXMPTAM SLIB variable 415
AJXMPTAS SLIB variable 416
AJXMPTHX SLIB variable 416
AJXMPUNI SLIB variable 416
AJXMISSID SLIB variable 447
AJXNDOPT SLIB variable 403
AJXNOSCR SLIB variable 406
AJXOBJT SLIB variable 448
AJXOBNAM SLIB variable 447
AJXOBNOD SLIB variable 447
AJXOBTL SLIB variable 447
AJXOCLN SLIB variable 412
AJXODEV SLIB variable 434
AJXODMSG SLIB variable 406
AJXODS44 symbolic variable 442
AJXODWNM SLIB variable 405
AJXOHMRC SLIB variable 412
AJXOIDSN SLIB variable 413
AJXOLGNM SLIB variable 415
AJXOTDSN SLIB variable 435
AJXOWLD SLIB variable 397
AJXOWLM SLIB variable 397
AJXOPLT SLIB variable 423
AJXPARTSLIB variable
  DSNUM symbolic variable 443
  LDSNUM symbolic variable 446
  LPART symbolic variable 446
  PA symbolic variable 448
  PART symbolic variable 448
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| AJXPB SLIB variable 448 | AJXR1EX SLIB variable 426 |
| AJXPB14 SLIB variable 424 | AJXR1MCL SLIB variable 426 |
| AJXPBST SLIB variable 424 | AJXR1PRF SLIB variable 426 |
| AJXPCHE SLIB variable 419 | AJXR1RIP SLIB variable 426 |
| AJXPCPS SLIB variable 420 | AJXR1SCL SLIB variable 426 |
| AJXPCPY SLIB variable 420 | AJXR1SF SLIB variable 427 |
| AJXPCST SLIB variable 420 | AJXR1TAD SLIB variable 426 |
| AJXPDOPT SLIB variable 401 | AJXR1TAM SLIB variable 426 |
| AJPDS1 SLIB variable 423 | AJXR1TAS SLIB variable 427 |
| AJXPDSN SLIB variable 423 | AJXR1THX SLIB variable 427 |
| AJXPDS1 SLIB variable 423 | AJXR2ALU SLIB variable 428 |
| AJXPFE1 SLIB variable 425 | AJXR2DC1 SLIB variable 427 |
| AJXPFRI SLIB variable 448 | AJXR2EX SLIB variable 427 |
| AJXPICM SLIB variable 424 | AJXR2MCL SLIB variable 427 |
| AJXPIS1 SLIB variable 424 | AJXR2PRF SLIB variable 427 |
| AJXPIST SLIB variable 422 | AJXR2PS SLIB variable 427 |
| AJXPKST SLIB variable 420 | AJXR2RP SLIB variable 428 |
| AJXPLOD SLIB variable 420 | AJXR2SCL SLIB variable 428 |
| AJXPLST SLIB variable 421 | AJXR2SF SLIB variable 428 |
| AJXPMNI SLIB variable 395 | AJXR2SS SLIB variable 428 |
| AJXPMXI SLIB variable 395 | AJXR2STK SLIB variable 428 |
| AJXPODATE edit macro 134 | AJXR2TAD SLIB variable 427 |
| AJXPOFER message file 136 | AJXR2TAM SLIB variable 427 |
| AJXPOFIN input stream 361 | AJXR2TAS SLIB variable 428 |
| AJXPOFIN keywords, list of 395 | AJXR2THX SLIB variable 428 |
| AJXPOFVL message file 136 | AJXR2UNI SLIB variable 428 |
| AJXQPST SLIB variable 422 | AJXRBF SLIB variable 428 |
| AJXPRCV SLIB variable 421 | AJXRBRC SLIB variable 429 |
| AJXPREO SLIB variable 421 | AJXRCPM SLIB variable 426 |
| AJXRPRSL SLIB variable 421 | AJXRCSL SLIB variable 426 |
| AJXPSSTV SLIB variable 424 | AJXRCUNI SLIB variable 427 |
| AJXPSST SLIB variable 421 | AJXRDOPT SLIB variable 402 |
| AJXPSTP SLIB variable 422 | AJXREGN SLIB variable 429 |
| AJXPSTT SLIB variable 421 | AJXRETPD SLIB variable 434 |
| AJXPRG SLIB variable 422 | AJXRLQ SLIB variable 448 |
| AJXPTSO SLIB variable 424 | AJXRPALU SLIB variable 431 |
| AJXPTST SLIB variable 424 | AJXRPMCL SLIB variable 430 |
| AJXPUCLSL SLIB variable 425 | AJXRPMCL SLIB variable 430 |
| AJXPDCL SLIB variable 425 | AJXRPPPRF SLIB variable 430 |
| AJXPEX SLIB variable 425 | AJXRPPS SLIB variable 430 |
| AJXPLD SLIB variable 422 | AJXRPRP SLIB variable 430 |
| AJXPLMC SLIB variable 425 | AJXRPSCL SLIB variable 430 |
| AJXPUFL SLIB variable 425 | AJXRPTSL SLIB variable 430 |
| AJXPUFL SLIB variable 425 | AJXRPTAD SLIB variable 430 |
| AJXPUPE SLIB variable 425 | AJXRPTAM SLIB variable 430 |
| AJXPUPE SLIB variable 425 | AJXRPTAS SLIB variable 430 |
| AJXPUSS SLIB variable 426 | AJXRPTHX SLIB variable 431 |
| AJXPUSS SLIB variable 426 | AJXRSFL SLIB variable 431 |
| AJXPUST SLIB variable 422 | AJXRSEQ SLIB variable 448 |
| AJXPUUNI SLIB variable 426 | AJXRUNTP SLIB variable 449 |
| AJXPVST SLIB variable 421 | AJXSCPS SLIB variable 417 |
| AJXPVST SLIB variable 421 | AJXSCSS SLIB variable 417 |
| AJXWIRK SLIB variable 395 | AJXSCUNI SLIB variable 418 |
| AJXWIRK SLIB variable 395 | AJXSEQ SLIB variable 449 |
| AJXWST SLIB variable 423 | AJXSINC SLIB variable 449 |
| AJXPZST SLIB variable 420 | AJXSINC SLIB variable 452 |
| AJXR1DCL SLIB variable 426 | AJXSLD1 SLIB variable 405 |
AJXSLD2 SLIB variable 405
AJXSOALU SLIB variable 432
AJXSODCL SLIB variable 431
AJXSOEX SLIB variable 431
AJXSOMCL SLIB variable 432
AJXSOPRF SLIB variable 432
AJXSOPS SLIB variable 432
AJXSORP SLIB variable 432
AJXSOSCL SLIB variable 432
AJXSOSS SLIB variable 432
AJXSOTAD SLIB variable 431
AJXSOTAM SLIB variable 432
AJXSOTAS SLIB variable 432
AJXSOTHX SLIB variable 432
AJXSOUNI SLIB variable 432
AJXSPNAM SLIB variable
   SN symbolic variable 449
   SPNAME symbolic variable 449
   TSIX symbolic variable 450
AJXSRPS SLIB variable 437
AJXSRSS SLIB variable 437
AJXSRUNI SLIB variable 437
AJXSSID SLIB variable
   JSSID symbolic variable 446
   SSID symbolic variable 449
AJXSTAKA SLIB variable 398
AJXSTAKB SLIB variable 400
AJXSTAKC SLIB variable 417
AJXSTAKR SLIB variable 426
AJXSTAKU SLIB variable 437
AJXSTEPC SLIB variable 450
AJXSTEPN SLIB variable 449
AJXSUALU SLIB variable 434
AJXSUDCL SLIB variable 433
AJXSUEX SLIB variable 433
AJXSUMCL SLIB variable 433
AJXSUPRF SLIB variable 433
AJXSUPS SLIB variable 434
AJXSURP SLIB variable 434
AJXSUSCL SLIB variable 434
AJXSUSS SLIB variable 434
AJXSUTAD SLIB variable 433
AJXSUTAM SLIB variable 433
AJXSUTAS SLIB variable 434
AJXSUTHX SLIB variable 434
AJXSUUNI SLIB variable 434
AJXSWNUM SLIB variable 431
AJXSWP5 SLIB variable 431
AJXSWSC SLIB variable 433
AJXSWSS SLIB variable 431
AJXSWT SLIB variable 433
AJXSWUNI SLIB variable 431
AJXSYND SLIB variable 433
AJXSYSEX SLIB variable 433
AJXSYM SLIB variable 450
AJXSZAP SLIB variable 407
AJXSZAT SLIB variable 407
AJXSZTC SLIB variable 408
AJXSZTL SLIB variable 408
AJXTBCR SLIB variable 450
AJXTBCRE SLIB variable 450
AJXTBNAM SLIB variable 450
AJXTBNOD SLIB variable 450
AJXTIMEP SLIB variable 435
AJXTRTCH SLIB variable 435
AJTXTS SLIB variable 450
AJXTSCR SLIB variable 450
AJXTSSID SLIB variable 450
AJXTU1 SLIB variable 435, 450
AJXTU2 SLIB variable 435, 451
AJXTU3 SLIB variable 435, 451
AJXU1ALU SLIB variable 438
AJXU1DCL SLIB variable 436
AJXU1EX SLIB variable 437
AJXU1MCL SLIB variable 437
AJXU1PRF SLIB variable 437
AJXU1RP SLIB variable 437
AJXU1SCL SLIB variable 437
AJXU1TAD SLIB variable 436
AJXU1TAS SLIB variable 437
AJXU1THX SLIB variable 437
AJXU2ALU SLIB variable 439
AJXU2DCL SLIB variable 438
AJXU2EX SLIB variable 438
AJXU2MCL SLIB variable 438
AJXU2PRF SLIB variable 438
AJXU2PS SLIB variable 438
AJXU2RP SLIB variable 438
AJXU2SCL SLIB variable 438
AJXU2SS SLIB variable 438
AJXU2STK SLIB variable 438
AJXU2TAD SLIB variable 438
AJXU2TAM SLIB variable 438
AJXU2TAS SLIB variable 438
AJXU2THX SLIB variable 439
AJXU2UNI SLIB variable 439
AJXUC SLIB variable 417
AJXUCMD SLIB variable 451
AJXUDOPT SLIB variable 451
AJXULLQ SLIB variable 451
AJXUPART SLIB variable 451
AJXUTID SLIB variable 452
AJXUVR1 SLIB variable 440, 452
AJXUVR2 SLIB variable 440, 452
AJXUVR3 SLIB variable 440, 452
AJXUVR4 SLIB variable 440, 452
AJXUVR5 SLIB variable 440, 452
AJXVC SLIB variable 435
AJXVCAT SLIB variable 452
AJXVDOPT SLIB variable 402
AJXWDCLA SLIB variable 441
AJXWKID SLIB variable
   JOBNAME symbolic variable 445
   WORKID8 symbolic variable 453
BINDDEPLOY command 459
BINDFAIL AEXIN keyword 384
BINDFAIL AJXPOTF keyword 399
block QQ command 382
BLRP data set 115
BLRP_DATACLASS AJXPOTF keyword 400
BLRP_DATACLASS_ALT AJXPOTF keyword 400
BLRP_EXPDT AJXPOTF keyword 400
BLRP_MGMTCLASS AJXPOTF keyword 400
BLRP_MGMTCLASS_ALT AJXPOTF keyword 400
BLRP_PREFIX AJXPOTF keyword 400
BLRP_PRIQTY AJXPOTF keyword 400
BLRP_RETPD AJXPOTF keyword 400
BLRP_SECQTY AJXPOTF keyword 400
BLRP_STACK AJXPOTF keyword 400
BLRP_STORCLASS AJXPOTF keyword 400
BLRP_STORCLASS_ALT AJXPOTF keyword 400
BLRP_THRESH AJXPOTF keyword 400
BLRP_UNIT AJXPOTF keyword 401
BLRP_UNIT_ALT AJXPOTF keyword 401
BMC Software solutions
Administrative Assistant for DB2 30
Database Administration for DB2 30
System Performance for DB2 30
BMC Software, contacting 2
BMC_CHECK_LOAD AJXPOTF keyword 401, 403
BMC_CHECK_OPTS AJXPOTF keyword 385, 401, 403
BMC_COPY_LOAD AJXPOTF keyword 401, 404
BMC_COPY_OPTS AJXPOTF keyword 385, 401, 405
BMC_LOAD_LOAD AJXPOTF keyword 401, 415
BMC_LOAD_OPTS AJXPOTF keyword 385, 401, 415
BMC_RECOVER_LOAD AJXPOTF keyword 402, 429
BMC_RECOVER_OPTS AJXPOTF keyword 387, 402, 429
BMC_REORG_LOAD AJXPOTF keyword 402, 429
BMC_REORG_OPTS AJXPOTF keyword 387, 402, 430
BMC_REORG_XBMID AJXPOTF keyword 403
BMC_UNLOAD_LOAD AJXPOTF keyword 403, 440
BMC_UNLOAD_OPTS AJXPOTF keyword 389, 403, 440
BMCACT_vr.Vvr_ATTR table 156
BMCACT_vr.Vvr_ATTR_VAL table 156
BMCACT_vr.Vvr_AUDIT_LOG table 156
BMCACT_vr.Vvr_CRS_VAL table 156
BMCACT_vr.Vvr_DLG table 156
BMCACT_vr.Vvr_DLG_ATTR table 156
BMCACT_vr.Vvr_EDITOR_USERS table 157
BMCACT_vr.Vvr_FCRS table 157
BMCACT_vr.Vvr_GMAP table 157
BMCACT_vr.Vvr_MSG table 157
BMCACT_vr.Vvr_RECOVERY_LOG table 157
BMCACT_vr.Vvr_SEARCH_VARS table 157
BMCACT_vr.Vvr_SESSION_LOG table 157
BMCACT_vr.Vvr_SQL_TABLE table 157
BMCACT_vr.Vvr_VIEW table 157
BMCADMF2 CLIST 63, 233
BMCASTR command 63
BMCCATS command 286, 462
BMCCHECK INDEX command 286, 461
BMCCHECK IX command 286, 461
BMCCHG command 63
BMCCHTS command 286, 462
BMCCOPY command 286, 462
BMCCOPY INDEX command 286, 462
BMCCOPY IX command 286, 462
BMCCP symbolic variable 442
BMCDASD command 63
BMCDASD CLIST 73, 162, 233
BMCEXPLORER (BMCEX) command 374
BMCEXPLORER command 462
BMCMLOAD command 286, 462
BMCREBUILD command 286, 462
BMCREBUILD INDEX command 286, 462
BMCREBUILD IX command 286, 462
BMCRECOVER command 286, 462
BMCRECOVER INDEX command 286, 462
BMCRECOVER IX command 286, 462
BMCREORG command 286, 462
BMCREORG INDEX command 286, 462
BMCREORG IX command 286, 462
BMCSSTATS command 286, 462
BMCSSTATS INDEX command 286, 462
BMCSSTATS IX command 286, 462
BMCMUHIST command 462
BMCMUNLOAD command 286, 462
BMCMUTIL command 462
BOPTS installation option 70
BP command 465
BP object type 39
BR command 150, 224, 466
BROWSE command 224, 456
browsing logs 351
buffer pool object type 39

C
C line command 473
C2 command 466
C2 object type 40, 183
CA command 466
CA object type 39
CANCEL command 76, 456, 470
CANCEL ISPF command 45
carriage return, hexadecimal format 45
CASCADE command 456
catalog access
   -I indicator (indirect) 48, 163
   -R indicator (direct) 48, 163
   -S indicator (server connection) 48
setting 35
   switching between direct and indirect 163
catalog indirection 163
CATALOG MANAGER
commands, list of 455
plans, list of 69
product code 69
switches 95, 175
tables 156
catalog statistics 200
CATALOGHELP command 456
CATAUDIT AEXIN keyword 384
CATAUDIT installation option 384
CATDOPT AEXIN keyword 384
CATRECOV installation option 384
CATRECOVER AEXIN keyword 384, 386
CATSTATS command 200, 465
CATUTIL AEXIN keyword 384
CD command 466
CD object type 39, 184
CHANGE command 470
Character Field Max Width option field 84
columns
nonprintable 44
nonviewable 44
check 2 object type 40
CHECK command 287, 462
check constraint object type 184
check dependent object type 39, 184
CHECK INDEX command 287, 462
CHECK IX command 287, 462
cHECK object type 39, 183
CHECK+_LOAD AJXPOFIN keyword 401, 403
CHECKDOPT AJXPOFIN keyword 401, 403
CHECKOPT AEXIN keyword 385
CHGMAN_LOAD AJXPOFIN keyword 403
CHKD command 287, 462
CHKI command 287, 462
Cl command 466
Cl object type 39
CK command 466
CK object type 39, 184
CL command 466
CL object type 39
cleanup, data set 108
CLEANUP RC AJXPOFIN keyword 404
CLIPBOARD command 317, 456
CLIST
BMCADMIF2 63, 233
BMCDB2 73, 162, 233
sample 152
variables 345
writing user commands 152
CLIST command 456
CMD command 456
CMD_DISABLE command equivalent 150
CNTL_DATACLASS AJXPOFIN keyword 404
CNTL_EXPDT AJXPOFIN keyword 404
CNTL_MGMTCLASS AJXPOFIN keyword 404
CNTL_PREFIX AJXPOFIN keyword 404
CNTL_PRIQTY AJXPOFIN keyword 404
CNTL_RETPD AJXPOFIN keyword 404
CNTL_SECQTY AJXPOFIN keyword 404
CNTL_STORCLASS AJXPOFIN keyword 404
CNTL_UNIT AJXPOFIN keyword 404
CO command 466
CO object type 39, 181, 184
collating sequence 198
collection object type 39
COLSTATS command 465
column authorization object type 39
column label object type 39
column object type 39, 184
combined lists
command format 180
excluding objects 182
generating 181
valid source objects 180
COMMAND (CMD) command 56
command name (exccomnd) 153
command program, writing 153
command recognition character 37
command reference 56
commands
abbreviating 58
adding 148
CATALOG MANAGER, list of 455
changing functionality 149
command-line, list of 470
data browsing 224
data browsing, list of 470
data editing 231
data editing, list of 470
disabling 148
disabling the commands table 148
erasing leftover characters 59
executing 56
Fast Path Navigation 63
including multiple objects 59
invoking BMC Software utilities 286
invoking IBM DB2 utilities 287
ISPF 45
issuing from Cmd column 51, 59
issuing from Command line 37
issuing similar 58
line, list of 473
list, list of 465
listing 56
omitting parameters 58
parameters 58
replacing format 148
shortcuts 60
typing 59
user-written 150
utility list, list of 464
utility, list of 461
commands (continued)
    valid for object list 58
    Wait-for-Enter (WFE) 59
COMMANDS command 456
commands table
    $ACTCMD macro 145
    ACTCOMND member 143
    adding user-written commands 150
    log option (excclog) 153
    modifying 148
    retaining from previous release 151
    REXX entry 145
    syntax and parameters 145
    tailoring 334
COMMENT command 459
common area, viewing 67
compiling and linking user commands table 148
compiling SLIBs 368
Confirm SQL panel
    for extended SQL processing 219
    setting default values 90
    working with SQL 212
CONNECT (CON) command
    attaching to SSID 161
    connecting to specified SSID 166, 167, 173
    connect and attach functions, defined 159
    CONNECT command 459
    CONNECT RESET command 164
    connecting to a specified SSID 166
    connecting to an SSID, using a saved connection 172
    Connections Table 173
constraint dependent object type 39, 184
CONTAB command 173
control characters, using in object names 44
conventions, documentation 20
COPY command 287, 462, 470
COPY INDEX command 287, 462
COPY IX command 287, 462
COPY+ LOAD AJXPOFIN keyword 401, 404
COPYAUTHS command 309, 317, 456
COPYDOPT AJXPOFIN keyword 401, 405
copying data
    Copy Table Rows option 234
    data compatibility 237
    overview 237
    using COPY command 237
    using Copy Table Rows option 241
    copying user privileges by privilege type 317
    COPYOPT AEXIN keyword 385
    COPYTOCOPY command 287, 462
    COPYTOCOPY INDEX command 287, 462
    COPYTOCOPY IX command 287, 462
    COUNT command 200, 459
    CP command 466
    CP object type 39, 184
    CP object type, search qualifier 44
    CR symbolic variable 442
CREATE command 459
creating an object hierarchy 268
creating and editing referential and unique constraints 260
creating objects, using DDL commands 267
creating tables
    copying table design 254
    creating and editing constraints 260
    defining columns 256
    using an existing object 254
CRS option with session profiles 346
CRS_VAL table 156
customer support 3
CUSTOMIZE (CU) command 335, 336, 343
CUSTOMIZE command 456
customizing CATALOG MANAGER
    authorization 334
    creating a session profile 340
    CRS option 346
    editing a tailored commands table 344
    initial list filters 341
    Primary Menu 336, 343
    types of customization 333
customizing object list display 196
CUT command 316
CX command 466
CX object type 39
CPA command 466
CPA object type 40
CPT command 466
CPT object type 40

D
D (DELETE) line command 473
D (DESCRIBE/SELECT) command 204
D command 456, 464
DA symbolic variable 442
DA TEMPLATE descriptor variable 442
DASD.DOPT AJXPOFIN keyword 405
DASD_LOAD AJXPOFIN keyword 405
DASDDOPT AEXIN keyword 385
data browsing commands, list of 470
data editing and browsing
    creating and editing data 234
    data locking 234
    invoking 224, 231
    no lock parameter 233
    row lock parameter 233
    setting options 231
    table lock parameter 232
data editing commands, list of 470
data locking 234
data set information, obtaining on a remote SSID 168
data set object type 40
data set sizing 362
Index

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

data sets
$AJXDOC 362
cleanup 108
DATAWORK 112
deleting automatically 108
discard 108
error 108
LOGSORT 112
map 108
naming conventions 88
options 89
permanent work 114
prefixes 116
punch 108
resolving names 117
sample VSAM object 107
setting default processing options 114
setting options 88
sizing 106
SORTOnnn 109
SORTOUT 109, 115
SORTPnnn 109
SORTWORK 112
SUTnnn 109
SYSCOPY 115
SYSDISC 115
SYSERR 115
SYSMAP 115
SYSPUNCH 115
SYSREC 115
SYSUT 115
SYSUTnnn 109
unload 108
work 108
WRKnnn 109
data type object type 40, 184
DATA_PACKER_LOAD AJXPOFIN keyword 405
Database Administration for DB2 30
database object type 38, 181
DATASETSIZING AJXPOFIN keyword 405
data type object type 40, 184
DATASETSIZING AJXPOFIN keyword 405
DATAWK_NBR AJXPOFIN keyword 405
DATAWK_UNIT AJXPOFIN keyword 405
DATWORK data set 112
DATE OUTPUT descriptor variable
  DATE symbolic variable 442
  DT symbolic variable 443
  JYMD symbolic variable 446
  YMD symbolic variable 453
DATE symbolic variable 442
DATE TEMPLATE descriptor variable
  DATE symbolic variable 442
  JYMD symbolic variable 446
  YMD symbolic variable 453
DATEJ symbolic variable 442
DAY OUTPUT descriptor variable 442
DAY symbolic variable 442
DAY TEMPLATE descriptor variable 442
DB command 466
DB object type 38, 178, 181, 184
DB OUTPUT descriptor variable
  CR symbolic variable 442
  DB symbolic variable 442
  DBNAME symbolic variable 442
  IXCR symbolic variable 444
  TBCR symbolic variable 450
  TBCRE symbolic variable 450
  VCAT symbolic variable 452
DB symbolic variable 442
DB TEMPLATE descriptor variable
  CR symbolic variable 442
  DB symbolic variable 442
  DBNAME symbolic variable 442
  IXCR symbolic variable 444
  TBCR symbolic variable 450
  TBCRE symbolic variable 450
  VCAT symbolic variable 452
DB..IS TEMPLATE descriptor variable 444
DB..SN TEMPLATE descriptor variable 447
DB..TS TEMPLATE descriptor variable 450
DB.TS OUTPUT descriptor variable
  IXNODE symbolic variable 444
  OBNAME symbolic variable 447
  OBNODE symbolic variable 447
  TBNODE symbolic variable 450
DB2
  initialization parameters, viewing 67
  plan name (exclplan) 153
  security 70
  special registers, viewing 67
  subsystem attached to (excssid) 153
DB2 Attach
  attach and connect, defined 159
  Call Attach Facility (CAF) 160
  default attachment, restoring 164
  from a connection server list 162
  requirements 160
  to a specified SSID 161
DB2 commands, issuing on a remote SSID 169
DB2 Connect
  change access panels 173
  Distributed Data Facility (DDF) 164
  requirements 164
  saved connections 170
  specified SSID 166
  using saved connection 172
DB2 DSN6 macros 67
DB2COMMAND command 459
DB2EXIT AJXPOFIN keyword 405
DB2-identifiers 174
DB2LOAD AJXPOFIN keyword 405
DB2STMSGS AEXIN keyword 385
DB2V2 symbolic variable 442
DB2V3 symbolic variable 443
DBNAME symbolic variable 442

Index  493
DBRM object type 39, 184
DBRMs, explaining with SQL Explorer 375
DCL command 321, 456
DCLGEN command 460
DDD symbolic variable 443
DDL BATCH processing 207
DDL command 267, 456
DDNAME symbolic variable 443
DDOPT symbolic variable 443
DDSEQ symbolic variable 443
debugging, setting default processing options 121
DEBUGUNLD AEXIN keyword 385
Decimal Point option field 83
DEF_GDG_BASE AJXPOFIN keyword 405
DEF_GDG_LIMIT AJXPOFIN keyword 405
DEF_GDG_NOSCR AJXPOFIN keyword 406
DEF_GDG2_LIMIT AJXPOFIN keyword 406
Default SQLID option field 77
defaults, setting product options
data sets 114
debugging 121
generation data groups 120
JCL jobcard 101
JCL static data sets 105
JCL STEPLIBs 104
LISTDEF data set 129
product options file 132
SORTWORK data sets 112	
tapes 109
TEMPLATE data set 129
utility module names 124
defaults, setting user options
basic options 79
general 72, 76
general options 82
non-worklist JCL 127
object use options 85
online reorgs 126
options data set 77
SHRLEVEL CHANGE 126
user variables 130
defining an options data set 77
defining table columns 256
DELETE (DEL) command 190
DELETE command 460
deleting, data sets automatically 108
delimited identifiers 83
dependency object type 40
DES command 204, 456
DESCRIBE BATCH processing 207
DESCRIBE command 202, 204, 374, 376, 456
DESCRIBE options 98
DEST command 465
DESTATISTICS (DEST) command 204
DESTATISTICS command 465
development aids 152
DIAG_MSGCLASS AJXPOFIN keyword 406
DIS command 460
disabling commands 148
DISC_DATACLASS AJXPOFIN keyword 406
DISC_DATACLASS_ALT AJXPOFIN keyword 406
DISC_EXPDT AJXPOFIN keyword 406
DISC_MGMTCLASS AJXPOFIN keyword 406
DISC_MGMTCLASS_ALT AJXPOFIN keyword 406
DISC_PREFIX AJXPOFIN keyword 406
DISC_PRIQTY AJXPOFIN keyword 406
DISC_RETFPOFIND keyword 406
DISC_SECQTY AJXPOFIN keyword 407
DISC_STORCLASS AJXPOFIN keyword 407
DISC_STORCLASS_ALT AJXPOFIN keyword 407
DISC_THRESHAJXPOFIN keyword 407
DISC_UNIT AJXPOFIN keyword 407
discard data set 108
DISDATABASE command 460
DISP_ALLOW_POPUP AJXPOFIN keyword 407
DISP_AUTO_TAB AJXPOFIN keyword 407
DISP_LOCATION AJXPOFIN keyword 408
DISP_OMIT_CHAR AJXPOFIN keyword 408
DISP_STATS AJXPOFIN keyword 408
DISPLAY command 460
DISPLAY DB2 command 207
displaying truncated utility profile ID 293
DISTHREAD command 460
distinct type object type 40, 184
distinct type object type 40, 184
documentation
field-level Help 19, 46
online Help 19
related publications 20
DOPTS command 72, 456
DOWN command 470
DOWN ISPF command 45
DP command 466
DP object type 40
*DROP marker 278
DROP and DROP RECOVERY functions
batch processing 283
excluding indexes from recovery 281
objects excluded from recovery 281
recovering multiple indexes 284
table spaces 274, 279
DROP command 274, 460
DROP IS switch 278
DROP NOT DONE message 278
Drop Recovery Log
actions recorded 355
recording simulated DROP 274
dropping pending changes 272
dropping table spaces 272
DROP command 279
DROP_NOIC POF keyword 386, 408
DROPRECOVERY command 457
DS command 466
DS object type 40
DSN1COMP command 287, 462
DSN1COPY command 287, 462
DSN1COPY utility 283
DSNCHECK44 AJXPOFIN keyword 409
DSNEXT library 67
DSNHDECP parameter 67
dynamic SQL 183
DYNWORKUNIT AEXIN keyword 385

dropping pending changes 272
dropping table spaces 272
DROP command 279
DROP_NOIC POF keyword 386, 408
DROPRECOVERY command 457
DS command 466
DS object type 40
DSN1COMP command 287, 462
DSN1COPY command 287, 462
DSN1COPY utility 283
dynamic SQL 183
DYNWORKUNIT AEXIN keyword 385

dropping pending changes 272
dropping table spaces 272
DROP command 279
DROP_NOIC POF keyword 386, 408
DROPRECOVERY command 457
DS command 466
DS object type 40
DSN1COMP command 287, 462
DSN1COPY command 287, 462
DSN1COPY utility 283
DSNCHECK44 AJXPOFIN keyword 409
DSNEXT library 67
DSNHDECP parameter 67
dynamic SQL 183
DYNWORKUNIT AEXIN keyword 385

E

e command 231
EBDIC collating sequence 198
ED command 150, 231, 301, 464, 466
EDIT command 231, 457, 470
editing the commands table 148
EDITOR_USERS table 157
electronic documentation 19
EN command 466
EN object type 40
END command, creating and editing tables 470
END ISPF command 45
enquiry character, hexadecimal format 45
ENTER ISPF command 45
ENV AEXIN keyword 385
ENVIRONMENT (ENVI) command 67
ENVIRONMENT command 457
environment variables object type 40
ERR_DATACLASS AJXPOFIN keyword 409
ERR_DATACLASS_ALT AJXPOFIN keyword 409
ERR_EXPDT AJXPOFIN keyword 409
ERR_MGMTCLASS AJXPOFIN keyword 409
ERR_MGMTCLASS_ALT AJXPOFIN keyword 409
ERR_PREFIX AJXPOFIN keyword 409
ERR_PRIQTY AJXPOFIN keyword 409
ERR_RETPD AJXPOFIN keyword 409
ERR_SECQTY AJXPOFIN keyword 410
ERR_STORCLASS AJXPOFIN keyword 410
ERR_STORCLASS_ALT AJXPOFIN keyword 410
ERR_THRESH AJXPOFIN keyword 410
ERR_UNIT AJXPOFIN keyword 410
ERR_UNIT_ALT AJXPOFIN keyword 410
error data set 108
excclog parameter 153
exccnum parameter 153
exccobjc parameter 153, 154
exccobjl parameter 153, 154
exccomnd parameter 153
exccsrc parameter 153
EXCEPTIONS command 463
EXCHANGE command 460
excluding objects from processing 62, 281
execplan parameter 153
excrc parameter 153
excsesid parameter 153
excssid parameter 153
extcoid parameter 153
EXEC command 287, 463
EXEC_LOAD AJXPOFIN keyword 410
Execution plans, using in CATALOG MANAGER 70
execution, worklist
  cleanup job step for data sets 108
delleting data sets automatically 108
using Execution plans 70
EXPLAIN command 457
extended SQL processing 219
external hexadecimal format 44

F

F1 (HELP) 45
F10 (LEFT) 45, 472
F11 (RIGHT) 46, 472
F2 (SPLIT) 46
F3 (END) 45, 470
F4 (RFIND) 472
F7 (UP) 46
F8 (DOWN) 45
F9 (SWAP) 46
Fast Path Navigation 63, 233
FCMD symbolic variable 443
FCPY_DATACLASS AJXPOFIN keyword 410
FCPY_EXPDT AJXPOFIN keyword 410
FCPY_MGMTCLASS AJXPOFIN keyword 410
FCPY_PREFIX AJXPOFIN keyword 410
FCPY_PRIQTY AJXPOFIN keyword 411
FCPY_RETPD AJXPOFIN keyword 411
FCPY_SECQTY AJXPOFIN keyword 411
FCPY_STORCLASS AJXPOFIN keyword 411
FCPY_SUPPRESS_SUFF AJXPOFIN keyword 411
FCPY_UNIT AJXPOFIN keyword 411
FCRS table 157
file tailoring 367
FILT_DATACLASS AJXPOFIN keyword 411
FILT_EXPDT AJXPOFIN keyword 411
FILT_MGMTCLASS AJXPOFIN keyword 411
FILT_PREFIX AJXPOFIN keyword 411
FILT_PRIQTY AJXPOFIN keyword 412
FILT_RETPD AJXPOFIN keyword 412
FILT_SECQTY AJXPOFIN keyword 412
FILT_STORCLASS AJXPOFIN keyword 412
FILT_UNIT AJXPOFIN keyword 412
FIND command 471
FK command 466
FK object type 40
FLOW AEXIN keyword 385
FN command 466
FN object type 40, 184
FO command 466
FO object type 40
foreign key object type 40
foreign key, data set sizing 364
form feed character, hexadecimal format 45
FP command 466
FP object type 40, 184
FREE ALL ALL command 460
FREE command 71, 460
FREE INACT command 460
FREE INACTIVE command 460
FREEZE command 471
FS command 466
FS object type 40, 184
FUNC parameter 154
function routine object type 40
function to execute on return (exrc) 153

G

GDG (generation data group)
setting default processing options 120
specifying 116
symbolic variable 117
using for data sets 116
GDG symbolic variable 116, 443
GDG_MODEL AJXPOFIN keyword 412
general options, setting 82
generating batch JCL 210
generating JCL by batch processing 206
generation data group. See GDG
GET command 374, 377, 457
GMAP table 157
GRANT command 460
granting table privileges 307
GRPNM symbolic variable 443

H

HASHFAIL AEXIN keyword 385
HASHFAIL AJXPOFIN keyword 412
HASHWARNRC AJXPOFIN keyword 412
HC command 90, 457
HDDL
   AUTH switch 269
   batch processing 206
   including GRANT 269
HDDL command 268, 457
HDDL CONCAT command 268
HDDL output data set 209
HDESC command 206
HDESC (HDESC) command 206
HDESCRIBE (HDESC) command 206
HDESCRIBE BATCH processing 207
HDESCRIBE command 206, 457
HELP ISPF command 45
Help, online 19
hexadecimal format, null character 45
Hierarchical GRANT (HGRANT) command 312
HISTORY command 465
HM symbolic variable 443
HMS symbolic variable 443
HOP symbolic variable 444
HO TEMPLATE descriptor variable 444
HO.MI TEMPLATE descriptor variable
   HM symbolic variable 443
   TIME4 symbolic variable 450
host variables
   in SQL model statements 215
   long values 191
testing 219
   used in SEARCH 190
   using quotation marks 192
   valid values in DML statements 192
HOUR OUTPUT descriptor variable 444
HOUR symbolic variable 444
HOUR.MINUTE OUTPUT descriptor variable
   HM symbolic variable 443
   TIME4 symbolic variable 450
HPRINT command 457

I

I line command 473
IBM DB2 utilities available in CATALOG MANAGER 287
IC command 466
IC object type 40, 181, 184
IC symbolic variable 444
IC TEMPLATE descriptor variable
   IC symbolic variable 444
   ICTYPE symbolic variable 444
   JOBTYP symbolic variable 445
   OBJT symbolic variable 447
   OBJTYP symbolic variable 445
   RTYPE symbolic variable 448
   RUNTYP symbolic variable 449
   TYPE symbolic variable 451
ICTYPE OUTPUT descriptor variable 444
ICTYPE symbolic variable 444
ICTYPE TEMPLATE descriptor variable 444
identity column object type 41
identity columns 259
IEFUSI exit 71
IL command 466
IL object type 40
IM (data editing) command 472
IM command 466
IM object type 40, 181
image copy object type 40, 184
IMPORT command 457
IN command 466
IN object type 40, 184
including SEARCH in BATCH jobs 194
index and table space partition object type 41
index mixed object type 40
index object type 39, 184
index space object type 40, 184
index space partition object type 40, 44, 184
index space statistics object type 40
indexes, recovering multiple 284
INDEXSTATS command 465
initial list filters
creating 341
retrieving 345
initial POF, refreshing 135
input streams 72
INSERT command 460
installation requirements 34
internal format 44
INVALID COMMAND message 382
IOALOAD1 AJXPOFIN keyword 412
IOALOAD2 AJXPOFIN keyword 412
IP command 466
IP list object type 40
IP name object type 40, 184
IP object type 40, 44, 184
IS command 466
IS object type 40, 184
IS symbolic variable 444
IS TEMPLATE descriptor variable 444
ISPF
AJXPODAT macro 134
commands 45
commands in CATALOG MANAGER 256
file tailoring 367
file tailoring for JCL generation 362
help 45
sharepool variables 152
variables 156
variables in data set names 88
ISS command 467
ISS object type 40
ITERATIONMODE AEXIN keyword 386
IX (data editing) command 472
IX command 467
IX object type 39, 181, 184
IX symbolic variable 444
IXCR symbolic variable 444
IXNAME symbolic variable 444
IXNODE symbolic variable 444
IXSPC symbolic variable 444
J
jar contents object type 40
jar object type 40
Java archive contents object type 184
Java archive object type 184
Java option object type 40
Java path object type 40
JB command 467
JB object type 40, 184
JC command 467
JC object type 40, 184
JCL Generation
data set sizing 363
ISPF file tailoring 362
product options 73
RUNSTATS utility 363
setting options 86
VSAM object sampling 363
JCL job card, setting default processing options 101
JCL job statement, defaults 102
JCL static data sets, setting default processing options 105
JCL STEPLIBs, setting default processing options 104
JCLCLEANUP AJXPOFIN keyword 412
JCLLIB AJXPOFIN keyword 412
JD symbolic variable 444
JDATE OUTPUT descriptor variable
DATEJ symbolic variable 442
JDATE symbolic variable 444
JU symbolic variable 446
JUL4Y symbolic variable 446
JULIAN symbolic variable 446
YYYDDD symbolic variable 453
YYYYDDD symbolic variable 453
JDATE symbolic variable 444
JDATE TEMPLATE descriptor variable
DATEJ symbolic variable 442
JDATE symbolic variable 444
JU symbolic variable 446
JUL4Y symbolic variable 446
JULIAN symbolic variable 446
YYYDDD symbolic variable 453
YYYYDDD symbolic variable 453
JDATE symbolic variable 444
JDAY OUTPUT descriptor variable
DDD symbolic variable 443
JDAY symbolic variable 444
JJULD symbolic variable 445
JDAY symbolic variable 444
JDAY TEMPLATE descriptor variable
   DDD symbolic variable 443
   JDAY symbolic variable 444
   JJULD symbolic variable 445
JDDN symbolic variable 444
JES3 AJXPOFIN keyword 412
JHMS symbolic variable 444
JJULD symbolic variable 445
JOB_INCLUDE_MEMBER AJXPOFIN keyword 412
JOBCHAR OUTPUT descriptor variable
   JOBCHAR symbolic variable 445
   JOBNAME symbolic variable 445
   JPCOD symbolic variable 445
   MEMBER symbolic variable 447
   MEMBR symbolic variable 447
   PGMR symbolic variable 448
   WKOWN symbolic variable 452
   WKOWNER symbolic variable 452
JOBNAME symbolic variable 445
JOBNAME TEMPLATE descriptor variable
   ATTACH symbolic variable 442
   DDOPT symbolic variable 443
   JDDN symbolic variable 444
   JOBCHAR symbolic variable 445
   JOBNAME symbolic variable 445
   JPCOD symbolic variable 445
   MEMBER symbolic variable 447
   MEMBR symbolic variable 447
   PGMR symbolic variable 448
   WKOWN symbolic variable 452
   WKOWNER symbolic variable 452
   ZACCTNUM symbolic variable 453
JOBTYP symbolic variable 445
JOIN command 457
JP command 467
JP object type 40
JPCOD symbolic variable 445
JQID symbolic variable 446
JS1 symbolic variable 445
JS2 symbolic variable 445
JS4 symbolic variable 446
JSSID symbolic variable 446
JT command 467
JT object type 40
JU symbolic variable 446
JUL4Y symbolic variable 446
JULIAN symbolic variable 446
JYMD symbolic variable 446

K

KC (data editing) command 472
KC command 467
KC object type 40
key column object type 40
key column use object type 184
key column user object type 40
key-target distribution history object type 42
key-target distribution object type 40
key-target distribution statistics history object type 42
key-target history object type 40
key-target object type 40
key-target statistics object type 40
KT command 467
KT object type 40
KTD command 467
KTD object type 40
KTDH command 467
KTDH object type 42
KTDS command 467
KTDS object type 42
KTH command 467
KTH object type 40
KTS command 467
KTS object type 40
KU command 467
KU object type 40, 184

L

LABEL command 460
last-used utility profile ID 299
LCAT command 469
LDSNUM symbolic variable 446
LEDIT command 457
LEFT command 472
LEFT ISPF command 45
LI MX command 467
LI symbolic variable 446
LI TEMPLATE descriptor variable 446
limit key object type 40
LINES AEXIN keyword 386
LIST action, default 38, 49
LIST command 467
list commands, list of 465
list object type (exccobjl) 153
LISTDEF control statements 300
LISTDEF data set 129
LISTDEF_DSN AJXPOFIN keyword 413
lists
   combined 180
   combined with SQL 215
   customizing displays 196
lists (continued)
   describing 201
   generating 47
   mixed lists 178
   printing 206
   using Quick-Search 192
   using SEARCH 183

 literal strings 44
 LK command 467
 LK object type 40
 LL object type 467
 LL object type 40, 184
 LL CLIB AJXPOFIN keyword 413
 LL CLIB2 AJXPOFIN keyword 413
 LL CLIB3 AJXPOFIN keyword 413
 LL CLIB4 AJXPOFIN keyword 413
 LL CLIB5 AJXPOFIN keyword 413
 LL LINCOLPFIN keyword 413
 LL LINCOLPFIN keyword 413
 LL LINCOLPFIN keyword 413
 LL LINK AJXPOFIN keyword 413
 LL LINK2 AJXPOFIN keyword 413
 LL LINK3 AJXPOFIN keyword 413
 LL LINK4 AJXPOFIN keyword 413
 LL LINK5 AJXPOFIN keyword 413
 LL MLIB AJXPOFIN keyword 413
 LL MLIB2 AJXPOFIN keyword 413
 LL MLIB3 AJXPOFIN keyword 413
 LL MLIB4 AJXPOFIN keyword 413
 LL MLIB5 AJXPOFIN keyword 413
 LL PLIB AJXPOFIN keyword 413
 LL PLIB2 AJXPOFIN keyword 414
 LL PLIB3 AJXPOFIN keyword 414
 LL PLIB4 AJXPOFIN keyword 414
 LL PLIB5 AJXPOFIN keyword 414
 LL PLIB AJXPOFIN keyword 414
 LL PLIB2 AJXPOFIN keyword 414
 LL PLIB3 AJXPOFIN keyword 414
 LL PLIB4 AJXPOFIN keyword 414
 LL PLIB5 AJXPOFIN keyword 414
 LL PLIB AJXPOFIN keyword 414
 LL PLIB2 AJXPOFIN keyword 414
 LL PLIB3 AJXPOFIN keyword 414
 LL PLIB4 AJXPOFIN keyword 414
 LL PLIB5 AJXPOFIN keyword 414
 LL PLIB AJXPOFIN keyword 414
 LL PLIB2 AJXPOFIN keyword 414
 LL PLIB3 AJXPOFIN keyword 414
 LL PLIB4 AJXPOFIN keyword 414
 LL PLIB5 AJXPOFIN keyword 414
 LL PLIB ALJXPOFIN keyword 414
 LL PLIB2 AJXPOFIN keyword 414
 LL PLIB3 AJXPOFIN keyword 414
 LL PLIB4 AJXPOFIN keyword 414
 LL PLIB5 AJXPOFIN keyword 414
 LL PLIB ALJXPOFIN keyword 414
 LL XML AJXPOFIN keyword 414
 LL XML2 AJXPOFIN keyword 414
 LL XML3 AJXPOFIN keyword 414
 LL XML4 AJXPOFIN keyword 414
 LL XML5 AJXPOFIN keyword 414
 LL Q AJXPOFIN keyword 415
 LL Q symbolic variable 446
 LM command 467
 LM object type 40, 184
 LO command 467
 LO list, connecting to a remote SSID 169
 LO object type 39, 184
 LOAD command 287, 463
 LOAD+ LOAD AJXPOFIN keyword 40, 415
 LOADDOPT AJXPOFIN keyword 40, 415
 LOADOPT AEXIN keyword 386
 LOB SYSREC data set 115
 LOBSTATS object type 41
 location list, connecting to a remote SSID 169
 location object type 39, 184
 lock contention during data editing 234
 locking options 63, 233
 LOCREM symbolic variable 446
 LOCREM TEMPLATE descriptor variable 446
 log text pointer (pointer) 154
 LOGRA command 457
 logs
   accessing 350
   administering audit logs 355
   administrative functions 350
   browsing 351
   DDL Audit Log 355
   Drop Recovery Log 355
   enable session 146
   purging 353
   Session Log 351
   types maintained 349
 LOGSORT data set 112
 LOGWK NBR AJXPOFIN keyword 415
 LOGWK UNIT AJXPOFIN keyword 415
 long names, truncating 84
 LPAR OUTPUT descriptor variable 446
 LPAR symbolic variable 446
 LR symbolic variable 446
 LR TEMPLATE descriptor variable 446
 LS command 467
 LS object type 40, 184
 LU command 467
 LU list object type 40, 184
 LU mode object type 40, 184
 LU mode select object type 40, 184
 LU name object type 40, 184
 LU object type 40, 184

 M
 M line command 473
 macros
   $ACTEXC 152, 153
   $ACTULOG 152, 153
   AJXPODAT 134
 MAINT command 354
 MAINTAIN command 350, 354, 457
 map data set 108
 MAP DATACLASS AJXPOFIN keyword 415
 MAP DATACLASS ALT AJXPOFIN keyword 415
 MAP EXPDT AJXPOFIN keyword 415
 MAP MGMTCLASS AJXPOFIN keyword 415
 MAP MGMTCLASS ALT AJXPOFIN keyword 415
 MAP PREFIX AJXPOFIN keyword 415
 MAP PRIQTY AJXPOFIN keyword 416
N

name fields in user-written commands 155
NAME parameter 154
native SQL procedure object type 41, 184
navigating CATALOG MANAGER 47
negative acknowledgegment, hexadecimal format 45
NOAPFOK AEXIN keyword 386
NOFAILNOIMAGECOPY AEXIN keyword 386
NOLOADCOMP AEXIN keyword 386
nonprintable characters 44
nonviewable characters 44
non-worklist JCL 127
NOSQLCOMP AEXIN keyword 386
NOSTARTOVER AEXIN keyword 386, 388
NOTIFYUNLD AEXIN keyword 387
NOWKIDREPLACE AEXIN keyword 387
NP command 468
NP object type 41, 184
null characters 44
number of objects option (exccnum) 153

O

OB command 468
OB object type 41
object lists
customizing display 196
ecluding objects 62
drom different source object types 54
drom multiple source objects 52
generating from Primary Menu 49
gordering columns 197
secondary lists 49
selecting objects 61
ordering by column 198
object role dependency object type 41
object type in command text (exccobjc) 153
object types
generating lists 38
passing in user-written commands 155
OBJT symbolic variable 447
OBJTYP symbolic variable 447
OBNAM symbolic variable 447
OBNOD symbolic variable 447
OBNOID symbolic variable 447
online command reference 56
online Help 19, 46
online reorgs 126
online schema changes object type 41
OPT command (data editing and browsing) 472
OPT parameter 154
options
changing 71
data editing and browsing 224
data set names 88, 89
data sets 114

mixed object lists
excluding objects 180
generating 179
valid source objects 178
mixed object type 40
MK command 467
MMDD symbolic variable 447
MO symbolic variable 447
MO TEMPLATE descriptor variable 447
models
creating new objects 59
creating tables 254
MODIFY command 287, 463
MODIFYRECOVERY command 287, 463
MODIFYSTATISTICS command 463
MODIFYSTATS command 287
MONTH OUTPUT descriptor variable 447
MONTH symbolic variable 447
MONTH TEMPLATE descriptor variable 447
MONTH.DAY OUTPUT descriptor variable 447
MONTH.DAY TEMPLATE descriptor variable 447
MORE command 235, 472
MQT command 467
MQT object type 40
MSG table 157
MSSID symbolic variable 447
multitasking, using tape stacking 111
MX command 179, 468
MX object type 40

map_retpd ajxpofin keyword 416
map_sectqy ajxpofin keyword 416
map_storclass ajxpofin keyword 416
map_storclass_alt ajxpofin keyword 416
map_unit ajxpofin keyword 416
map_unit_alt ajxpofin keyword 416
materialized query table object type 40
max_cyl ajxpofin keyword 416
max_priqty ajxpofin keyword 416
max_seqty ajxpofin keyword 416
max_unitcnt ajxpofin keyword 417
mddl command 269, 457
member symbolic variable 447
membr symbolic variable 447
memlimit ajxpofin keyword 417
memlimit system parameter 71
mergecopy command 287, 463
message files 136
message, invalid command 382
mi symbolic variable 447
mi template descriptor variable 447
minute output descriptor variable 447
minute symbolic variable 447
minute template descriptor variable 447
mixed object lists
excluding objects 180
generating 179
valid source objects 178
mixed object type 40
mk command 467
mmdd symbolic variable 447
mo symbolic variable 447
mo template descriptor variable 447
models
creating new objects 59
creating tables 254
modify command 287, 463
modifyrecovery command 287, 463
modifystatistics command 463
modifystats command 287
month output descriptor variable 447
month symbolic variable 447
month template descriptor variable 447
month.day output descriptor variable 447
month.day template descriptor variable 447
more command 235, 472
mqt command 467
mqt object type 40
msg table 157
mssid symbolic variable 447
multitasking, using tape stacking 111
mx command 179, 468
mx object type 40
## Options (continued)
- Debugging and display 121
- DESCRIBE 98
- General 82
- Generation data group 120
- Installation 72
- JCL Generation 86
- JCL job cards 101
- JCL static data sets 105
- JCL STEPLIBs 104
- LISTDEF data set 129
- Non-worklist JCL 127
- Online reorg 126
- Option switches 95, 175
- Panel attributes 94
- Product options file 132
- Refreshing 73
- Setting 71
- Setting values 76
- SHRLEVEL CHANGE 126
- SORTWORK data sets 112
- SQL command 90
- SQL SELECT 93
- Tapes 109
- TEMPLATE data set 129
- Used by products 75
- User 72
- User variables 130
- Utility module names 124
- OPTIONS command 457
- Options data set, defining 77
- ORDER command 458
- Ordinary identifiers 83
- ORTPARM_DSN AJXPOFIN keyword 417
- OS command 468
- OS object type 41
- OUTPUT descriptor variables, list of 442

## Parameters
- MEMLIMIT 71
- System 71
- Parsing for object name 154
- Parsing object types and names 154
- PART OUTPUT descriptor variable
  - BMCCP symbolic variable 442
  - PA symbolic variable 448
  - PART symbolic variable 448
  - PART4 symbolic variable 448
  - PART5 symbolic variable 448
  - UP symbolic variable 451
  - UPART symbolic variable 451
- PART symbolic variable 448
- PART TEMPLATE descriptor variable
  - BMCCP symbolic variable 442
  - DSNUM symbolic variable 443
  - LPART symbolic variable 446
  - PART symbolic variable 448
  - PART4 symbolic variable 448
  - PART5 symbolic variable 448
  - UP symbolic variable 451
  - UPART symbolic variable 451
- PART4 symbolic variable 448
- PART5 symbolic variable 448
- PASTE command 317
- PB symbolic variable 448
- PB TEMPLATE descriptor variable 448
- PCPY1_DATACLASS AJXPOFIN keyword 417
- PCPY1_DATACLASS_ALT AJXPOFIN keyword 417
- PCPY1_EXPDT AJXPOFIN keyword 417
- PCPY1_MGMTCLASS AJXPOFIN keyword 417
- PCPY1_MGMTCLASS_ALT AJXPOFIN keyword 417
- PCPY1_PRIQTY AJXPOFIN keyword 417
- PCPY1_RETPD AJXPOFIN keyword 417
- PCPY1_SECQTY AJXPOFIN keyword 417
- PCPY1_STACK AJXPOFIN keyword 417
- PCPY1_STORCLASS AJXPOFIN keyword 418
- PCPY1_STORCLASS_ALT AJXPOFIN keyword 418
- PCPY1_SUPPRESS_SUFF AJXPOFIN keyword 418
- PCPY1_THRESH AJXPOFIN keyword 418
- PCPY1_UNIT AJXPOFIN keyword 418
- PCPY1_UNIT_ALT AJXPOFIN keyword 418
- PCPY2_DATACLASS AJXPOFIN keyword 418
- PCPY2_EXPDT AJXPOFIN keyword 418
- PCPY2_MGMTCLASS AJXPOFIN keyword 418
- PCPY2_MGMTCLASS_ALT AJXPOFIN keyword 418
- PCPY2_PRIQTY AJXPOFIN keyword 418
- PCPY2_RETPD AJXPOFIN keyword 419
- PCPY2_SECQTY AJXPOFIN keyword 419
- PCPY2_STACK AJXPOFIN keyword 419
- PCPY2_STORCLASS AJXPOFIN keyword 419
- PCPY2_STORCLASS_ALT AJXPOFIN keyword 419
- PCPY2_SUPPRESS_SUFF AJXPOFIN keyword 419

## Packages
- Explaining statements with SQL Explorer 376
- Explaining with SQL Explorer 375
- Use by plans 181

## P
- P command 464
- PA command 468
- PA object type 41
- PA symbolic variable 448
- PA TEMPLATE descriptor variable 448
- Package object type 39, 184
- Packages
  - Explaining statements with SQL Explorer 376
  - Explaining with SQL Explorer 375
  - Use by plans 181
- PACKIT command 71, 458
- Packlist object type 41
- Packlists 181
- Panel attribute options, setting 94
- Parameter object type 184
PCPY2_THRESH AJXPOFIN keyword 419
PCPY2_UNIT AJXPOFIN keyword 419
PCPY2_UNIT_ALT AJXPOFIN keyword 419
PDD command 468
PDISTSTATS command 465
PEEK command 67
pending changes, dropping 272
percent character, hexadecimal format 45
permanent work data sets 115
*PERSIST option 82
PSHOW ISPF command 46
PG command 468
PG object type 39, 181, 184
PGC command 468
PGMR symbolic variable 448
PI command 181, 468
PI object type 41
PK command 468
PK object type 41
PL command 468
PL object type 39, 178, 181, 184
plan authorization object type 41
plan object type 39, 184
plans
  CATALOG MANAGER 69
    explaining with SQL Explorer 375
    manipulating 71
    package use 181
PM command 468
POF (product options file)
  adding steps to JCL 140
  AJXPODAT macro 134
  creating a user POF 132
  generating reports 136
  initial 73
  initializing 73
  obtaining list of TEMPLATEs or LISTDEFs 143
  overriding values in SLIBs 138
  POF Validation Report 136
  populating 73
  refreshing the initial POF 135
  reusing in a subsequent installation 137
  setting default processing options 132
  updating a user POF 133
  user 74
  using multiple POFs 134
  Variables Initialized with Default report 136
POF keywords, list of 395
POFDATE AJXPOFIN keyword 419
POFDATE POF keyword 74
POFDS installation option 73
pointer parameter 154
PR command 468
PR object type 39, 184
PR parameter in ACTEMAIN CLIST 345, 346
PREFIX symbolic variable 448
prefixes 116
PRI command 206
PRIBAC symbolic variable 448
PRIBAC TEMPLATE descriptor variable 448
primary key object type 41
PRINT CLOSE command 90
PRINT command 90, 206, 458
privileges
  copying by privilege type 317
  granting table privileges 307
to create objects 248
PRO command 458
PROC_BMCCHECK_NAME POF keyword 419
PROC_BMCCHECK_STEP POF keyword 420
PROC_BMCCOPY_NAME POF keyword 420
PROC_BMCCOPY_STEP POF keyword 420
PROC_BMCPPRS_NAME POF keyword 420
PROC_BMCPPRS_STEP POF keyword 420
PROC_BMCLOAD_NAME POF keyword 420
PROC_BMCLOAD_STEP POF keyword 421
PROC_BMCRECOVER_NAME POF keyword 421
PROC_BMCRECOVER_STEP POF keyword 421
PROC_BMCREORG_NAME POF keyword 421
PROC_BMCREORG_STEP POF keyword 421
PROC_BMCSTATS_NAME POF keyword 421
PROC_BMCSTATS_STEP POF keyword 421
PROC_BMCREGR_NAME POF keyword 421
PROC_BMCREGR_STEP POF keyword 421
PROC_BMCUNLOAD_NAME POF keyword 422
PROC_BMCUNLOAD_STEP POF keyword 422
PROC_BMCUPRS_NAME POF keyword 423
PROC_BMCUPRS_STEP POF keyword 423
PROC_DSN1COPY_NAME POF keyword 423
PROC_DSN1COPY_STEP POF keyword 423
PROC_DSNUTILB_NAME POF keyword 423
PROC_DSNUTILB_STEP POF keyword 423
PROC_GEN_SET_VAR POF keyword 424
PROC_IDCAMS_NAME POF keyword 424
PROC_IDCAMS_STEP POF keyword 424
PROC_IEFBR14_NAME POF keyword 424
PROC_IEFBR14_STEP POF keyword 424
PROC_TSO_NAME POF keyword 424
PROC_TSO_STEP POF keyword 424
PROC_USER_DEFINED POF keyword 425
PROC_USER_DEFINED POF keyword 425
PROC_USER_DEFINED POF keyword 425
PROC_USER_DEFINED POF keyword 424
PROC_USER_DEFINED POF keyword 424
procedure object type 184
product Help 19, 46
product options file. See POF
product support 3
PROFILE command 458
PROFILE command (session profiles) 338
PROFILE command (utility profiles) 300
PROFILE SAVE command 298
PROFILE SAVEAS command 292, 295, 297
PROFILE SET command 300
PROFILE.profileName ADDED message 337, 341, 342
PROFILE.profileName UPDATE message 344
PROFILES command 458
PT command 468
PT object type 41, 181
publications, related 20
punch data set 108
PUNCH_DATACLASS AJXPOFIN keyword 425
PUNCH_EXPDT AJXPOFIN keyword 425
PUNCH_MGMTCLASS AJXPOFIN keyword 425
PUNCH_PREFIX AJXPOFIN keyword 425
PUNCH_PRIQTY AJXPOFIN keyword 425
PUNCH_RETPD AJXPOFIN keyword 425
PUNCH_SECQTY AJXPOFIN keyword 426
PUNCH_UNIT AJXPOFIN keyword 426
PURGE COMPLETED message 354
purging logs 353
Q
QCONNECT command 460
QQ command 382
QRO command 468
QRP command 468
QRY command 468
QUAL parameter 154
Qualifier field
  IP object type 44
  object names 42
  objects with two-part names 44
  TP object type 44
  wildcard characters 42
quickname 172
quick-search
  in BATCH jobs 194
  using saved search variables 193
  WHERE clause 194
QUIESCE command 287, 463
R
R line command 473
RC parameter 154
RCHANGE command 472
RCPY1_DATACLASS AJXPOFIN keyword 426
RCPY1_DATACLASS_ALT AJXPOFIN keyword 426
RCPY1_EXPDT AJXPOFIN keyword 426
RCPY1_MGMTCLASS AJXPOFIN keyword 426
RCPY1_MGMTCLASS_ALT AJXPOFIN keyword 426
RCPY1_PREFIX AJXPOFIN keyword 426
RCPY1_PRIQTY AJXPOFIN keyword 426
RCPY1_RETPD AJXPOFIN keyword 426
RCPY1_SECQTY AJXPOFIN keyword 426
RCPY1_STACK AJXPOFIN keyword 426
RCPY1_STORCLASS AJXPOFIN keyword 426
RCPY1_STORCLASS_ALT AJXPOFIN keyword 427
RCPY1_SUPPRESS_SUFF AJXPOFIN keyword 427
RCPY1_THRESH AJXPOFIN keyword 427
RCPY1_UNIT AJXPOFIN keyword 427
RCPY1_UNIT_ALT AJXPOFIN keyword 427
RCPY2_DATACLASS AJXPOFIN keyword 427
RCPY2_DATACLASS_ALT AJXPOFIN keyword 427
RCPY2_EXPDT AJXPOFIN keyword 427
RCPY2_MGMTCLASS AJXPOFIN keyword 427
RCPY2_MGMTCLASS_ALT AJXPOFIN keyword 427
RCPY2_PREFIX AJXPOFIN keyword 427
RCPY2_PRIQTY AJXPOFIN keyword 427
RCPY2_RETPD AJXPOFIN keyword 428
RCPY2_SECQTY AJXPOFIN keyword 428
RCPY2_STACK AJXPOFIN keyword 428
RCPY2_STORCLASS AJXPOFIN keyword 428
RCPY2_STORCLASS_ALT AJXPOFIN keyword 428
RCPY2_SUPPRESS_SUFF AJXPOFIN keyword 428
RCPY2_THRESH AJXPOFIN keyword 428
RCPY2_UNIT AJXPOFIN keyword 428
RCPY2_UNIT_ALT AJXPOFIN keyword 428
RD command 468
RD object type 41
RE command 468
RE object type 41, 181, 184
real-time statistics. See RTS
REBIND command 71, 460
REBINDFAIL AEXIN keyword 387
REBINDFAIL AJXPOFIN keyword 428
REBINDRC AEXIN keyword 387
REBUILD command 287, 463
REBUILD INDEX command 287, 463
REBUILD IX command 287, 463
RECOVER command 283, 284, 287, 463
RECOVER INDEX command 287, 463
RECOVER DOPT AJXPOFIN keyword 402, 429
recovering objects 274
RECOVEROPT AEXIN keyword 387
RECOVERY_LOG table 157
referential integrity object type 41
REFRESH command 458, 461
refreshing user options 73
REGENERATE command 461
REGION AJXPOFIN keyword 429
register for return code (RC) 154
related publications 20
relation object type 41, 184
remote SSID, connecting 168, 169
RENAME command 461
reordering object list columns 197
REORG command 287, 463
REORG INDEX command 287, 463
REORG IX command 287, 463
REORG+_LOAD AJXPOFIN keyword 402, 429
REORG_MAPTAB AJXPOFIN keyword 429
REORGDOPORT AJXPOFIN keyword 402, 430
REORGLOPT AEXIN keyword 387
REPORT command 287, 463
REPORT INDEX command 287, 463
REPORT IX command 287, 463
reports, generating POF 136
REPT_DATACLASS AJXPOFIN keyword 430
REPT_DATACLASS_ALT AJXPOFIN keyword 430
REPT_EXPDT AJXPOFIN keyword 430
REPT_MGMTCLASS AJXPOFIN keyword 430
REPT_MGMTCLASS_ALT AJXPOFIN keyword 430
REPT_PRIQTY AJXPOFIN keyword 430
REPT_RETPD AJXPOFIN keyword 430
REPT_SECQTY AJXPOFIN keyword 430
REPT_STORCLASS AJXPOFIN keyword 430
REPT_STORCLASS_ALT AJXPOFIN keyword 430
REPT_THRESH AJXPOFIN keyword 431
REPT_UNIT AJXPOFIN keyword 431
REPT_UNIT_ALT AJXPOFIN keyword 431
RESET command 458
RESET subcommand 164
RESPONSES command 458
REVOKE command 323, 461
REXX commands table entry 145
RFIND command 472
RHLQ symbolic variable 448
RI command 468
RI object type 41
RIGHT command 472
RIGHT ISPF command 46
RETURN ISPF command 46
REVOKE command 323, 461
REXX commands table entry 145
RFIND command 472
RHLQ symbolic variable 448
RI command 468
RI object type 41
RIGHT command 472
RIGHT ISPF command 46
RO command 468
RO object type 41
roles object type 41
routine object type 184
routine option object type 40
routine parameter object type 40
routine source object type 40, 184
ROWID SYSREC data set 115, 118
ROWVIEW/ROW command 472
RP command 299, 464
RSEQ# symbolic variable 448
RTS (real-time statistics)
  index space 467
table space 469
RTYPE symbolic variable 448
RUNSTATS command 287, 463
RUNSTATS INDEX command 287, 463
RUNSTATS IX command 287, 463
RUNSTATS utility 363
RUNTIME_HLQ AJXPOFIN keyword 431
RUNTYP symbolic variable 449
S
S command 458
SAVE command 472
SC command 468
SC object type 41
SC symbolic variable 449
SC TEMPLATE descriptor variable 449
schema object type 41
SE command 468
SE object type 41, 184
SEARCH command 183, 458
SEARCH function
  associate with profile 188
  associate with user ID 188
  authorization required 183
  complex subqueries 194
  generating lists 185
  JOINed tables 195
  operator (Oper) variables 186
  retrieving saved variables 189
  saving variables 187
  setting values 186
  using host variables 190
  valid source objects 183
  WHERE clauses 187, 191
SEARCH_VARS table 157
SEC OUTPUT descriptor variable 449
SEC object type 449
SECOND OUTPUT descriptor variable 449
SECOND object type 449
SECOND TEMPLATE descriptor variable 449
SECOND secondary lists, generating from object lists 50
security, setting authorizations 72
SEE command 67, 461
SELECT command 461
SEQ OUTPUT descriptor variable
  DDPOT symbolic variable 443
  DDSEQ symbolic variable 443
  JDDN symbolic variable 444
  RSEQ# symbolic variable 448
  SEQ symbolic variable 449
  SEQ# symbolic variable 449
  SQ symbolic variable 449
SEQ symbolic variable 449
SEQ TEMPLATE descriptor variable
  DDSEQ symbolic variable 443
  RSEQ# symbolic variable 448
  SEQ symbolic variable 449
  SEQ# symbolic variable 449
SEQ symbolic variable 449
SEQ# symbolic variable 449
sequence object type 184
SESSION command 458
session logs
   actions captured 351
   browsing 351
   enabling 146
   purging 353
session profiles 333
   assigning to users 345
   authorization to create 334
   CLIST 345
   creating initial list filters 341
   CRS option 346
   CUSTOMIZE command 336
   customizing commands table 340
   customizing Primary Menu 336
   deleting 342
   displaying descriptions 339
   displaying list 338
   editing customized commands table 344
   editing customized Primary Menu 343
   PR parameter in ACTEMAIN CLIST 345, 346
   retrieving initial list filters 345
   SET PROFILE command 334
   TAILOR command 340
   types of customization 333
SESSION_LOG table 157
SET command 458
SG command 468
SG object type 38, 178, 181, 184
SHRLEVEL CHANGE 126
Simple Space Estimation (SSE) feature 46, 248
skeleton library compiler. See SLIB (skeleton library) compiler
skeleton library, overriding POF values 138
SLIB (skeleton library) compiler
   changing an ISPF skeleton 367
   compiling 366, 368
   ISPF file tailoring 367
   processing 369
   runtime report 369
   runtime report summary 371
   runtime unit 369
   testing changes 367
SLIB variables, list of 390, 442
SMFPRM member 71
SMS (Storage Management Subsystem) 119
SN symbolic variable 449
SN TEMPLATE descriptor variable 449, 450
solutions, BMC Software 30
SORT command 198, 458, 473
sorting object lists by column 198
SORTOnnn data set 109
SORTOUT data set 109, 115
SORTPnnn data set 109
SORTWK_NBR AJXPFOFIN keyword 431
SORTWK_PRIQTY AJXPFOFIN keyword 431
SORTWK_SECQTY AJXPFOFIN keyword 431
SORTWK_UNIT AJXPFOFIN keyword 431
SORTWORK data set 112
SPACE command 463
SPBXPRINT AEXIN keyword 387
specify log option (OPT) 154
SPLIT ISPF command 46
SPNAME symbolic variable 449
SQ symbolic variable 449
SQL TEMPLATE descriptor variable 449
SQL
   applying model statements 215
   copying external 215
   copying in SQL_Table 214
   creating new SQL from existing 215
   creating new SQL_Table entry 214
   escape character 83
   extended SQL processing 219
   generating 212
   SQL_Table 213
   SQL command 458
   SQL command options, setting 90
   SQL Explorer for DB2
      ACTPSS CLIST 374
      commands to access SQL 374
      integrating with CATALOG MANAGER 373
      SQLX edit macro 381
   SQL SELECT options, setting 93
   SQL_Table
      copying SQL 214
      creating new entry 214
      defined 213
      displaying 213
      explaining a statement with SQL Explorer 379
   SQL_TABLE table 157
   SQL_EXP_LOAD keyword 431
   SQLX edit macro 381
   SRTOUT_DATACLASS AJXPFOFIN keyword 431
   SRTOUT_DATACLASS_ALT AJXPFOFIN keyword 431
   SRTOUT_EXPDT AJXPFOFIN keyword 431
   SRTOUT_MGMTCLASS AJXPFOFIN keyword 432
   SRTOUT_MGMTCLASS_ALT AJXPFOFIN keyword 432
   SRTOUT_PFXAJXPFIN keyword 432
   SRTOUT_PRIQTY AJXPFOFIN keyword 432
   SRTOUT_RETPD AJXPFOFIN keyword 432
   SRTOUT_SECQTY AJXPFOFIN keyword 432
   SRTOUT_STORCLASS AJXPFOFIN keyword 432
   SRTOUT_STORCLASS_ALT AJXPFOFIN keyword 432
   SRTOUT_THRESH AJXPFOFIN keyword 432
   SRTOUT_UNIT AJXPFOFIN keyword 432
   SRTOUT_UNIT_ALT AJXPFOFIN keyword 432
   SS command 464
   SS symbolic variable 449
   SS TEMPLATE descriptor variable 449
   SSE command 248
   SSE ISPF command 46
<table>
<thead>
<tr>
<th>SSID</th>
<th>attaching 161 connecting 166 specifying at startup 35</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSID AEXIN keyword 388</td>
<td></td>
</tr>
<tr>
<td>SSID OUTPUT descriptor variable</td>
<td></td>
</tr>
<tr>
<td>GRPNM symbolic variable 443</td>
<td></td>
</tr>
<tr>
<td>JSSID symbolic variable 446</td>
<td></td>
</tr>
<tr>
<td>MSSID symbolic variable 447</td>
<td></td>
</tr>
<tr>
<td>SS symbolic variable 449</td>
<td></td>
</tr>
<tr>
<td>SSID symbolic variable 449</td>
<td></td>
</tr>
<tr>
<td>TSSID symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>SSID symbolic variable 449</td>
<td></td>
</tr>
<tr>
<td>SSID TEMPLATE descriptor variable</td>
<td></td>
</tr>
<tr>
<td>GRPNM symbolic variable 443</td>
<td></td>
</tr>
<tr>
<td>JSSID symbolic variable 446</td>
<td></td>
</tr>
<tr>
<td>MSSID symbolic variable 447</td>
<td></td>
</tr>
<tr>
<td>SS symbolic variable 449</td>
<td></td>
</tr>
<tr>
<td>TSSID symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>ST command 468</td>
<td></td>
</tr>
<tr>
<td>ST object type 39, 184</td>
<td></td>
</tr>
<tr>
<td>ST symbolic variable 449</td>
<td></td>
</tr>
<tr>
<td>START CLONE command 461</td>
<td></td>
</tr>
<tr>
<td>START command 284, 461</td>
<td></td>
</tr>
<tr>
<td>START DB2 command 207</td>
<td></td>
</tr>
<tr>
<td>STARTOVER AEXIN keyword 386, 388</td>
<td></td>
</tr>
<tr>
<td>static SQL 376, 379</td>
<td></td>
</tr>
<tr>
<td>STATS AEXIN keyword 388</td>
<td></td>
</tr>
<tr>
<td>STATS command 463</td>
<td></td>
</tr>
<tr>
<td>STATUS command 463</td>
<td></td>
</tr>
<tr>
<td>STEP# symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>STEP_INCLUDE_MEMBER AJXPOFIN keyword 432</td>
<td></td>
</tr>
<tr>
<td>STEPLIB libraries, setting default values 102</td>
<td></td>
</tr>
<tr>
<td>STEPN symbolic variable 449</td>
<td></td>
</tr>
<tr>
<td>STEPNNAME OUTPUT descriptor variable</td>
<td></td>
</tr>
<tr>
<td>STEPN symbolic variable 449</td>
<td></td>
</tr>
<tr>
<td>TU1 symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TU2 symbolic variable 451</td>
<td></td>
</tr>
<tr>
<td>TU3 symbolic variable 451</td>
<td></td>
</tr>
<tr>
<td>UDOPIT symbolic variable 451</td>
<td></td>
</tr>
<tr>
<td>WKOWN symbolic variable 452</td>
<td></td>
</tr>
<tr>
<td>WKOWNER symbolic variable 452</td>
<td></td>
</tr>
<tr>
<td>STEPNAME symbolic variable 449</td>
<td></td>
</tr>
<tr>
<td>STEPNNAME TEMPLATE descriptor variable</td>
<td></td>
</tr>
<tr>
<td>STEPN symbolic variable 449</td>
<td></td>
</tr>
<tr>
<td>TU1 symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TU2 symbolic variable 451</td>
<td></td>
</tr>
<tr>
<td>TU3 symbolic variable 451</td>
<td></td>
</tr>
<tr>
<td>UDOPIT symbolic variable 451</td>
<td></td>
</tr>
<tr>
<td>WKOWN symbolic variable 452</td>
<td></td>
</tr>
<tr>
<td>WKOWNER symbolic variable 452</td>
<td></td>
</tr>
<tr>
<td>stopgroup object type 38</td>
<td></td>
</tr>
<tr>
<td>STOP CLONE command 461</td>
<td></td>
</tr>
<tr>
<td>STOP command 282, 461</td>
<td></td>
</tr>
<tr>
<td>STOP DB2 command 207</td>
<td></td>
</tr>
<tr>
<td>STOPWAIT AEXIN keyword 388, 389, 433</td>
<td></td>
</tr>
<tr>
<td>STOPWAIT AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>STORAGE group object type 184</td>
<td></td>
</tr>
<tr>
<td>storage, virtual 71</td>
<td></td>
</tr>
<tr>
<td>stored procedure object type 39</td>
<td></td>
</tr>
<tr>
<td>stored procedures</td>
<td></td>
</tr>
<tr>
<td>ADMIN_COMMAND_DB2 164, 169, 170</td>
<td></td>
</tr>
<tr>
<td>ADMIN_DS_LIST 164</td>
<td></td>
</tr>
<tr>
<td>DSNWZP 460</td>
<td></td>
</tr>
<tr>
<td>DSNWZPT 164</td>
<td></td>
</tr>
<tr>
<td>STOPSPACE command 288, 463</td>
<td></td>
</tr>
<tr>
<td>string object type 39, 184</td>
<td></td>
</tr>
<tr>
<td>SU command 468</td>
<td></td>
</tr>
<tr>
<td>SU object type 39, 184</td>
<td></td>
</tr>
<tr>
<td>support, customer 3</td>
<td></td>
</tr>
<tr>
<td>SUPPRESS_COMMENTS AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>SYTrnn data set 109</td>
<td></td>
</tr>
<tr>
<td>SWAP ISPF command 46</td>
<td></td>
</tr>
<tr>
<td>switches, CATALOG MANAGER 95</td>
<td></td>
</tr>
<tr>
<td>switching between BMC Software products 63</td>
<td></td>
</tr>
<tr>
<td>SY command 468</td>
<td></td>
</tr>
<tr>
<td>SY object type 39, 178, 184</td>
<td></td>
</tr>
<tr>
<td>symbolic variables</td>
<td></td>
</tr>
<tr>
<td>GDG 117</td>
<td></td>
</tr>
<tr>
<td>list of 442</td>
<td></td>
</tr>
<tr>
<td>using in installation options module 441</td>
<td></td>
</tr>
<tr>
<td>using on product panels 441</td>
<td></td>
</tr>
<tr>
<td>SYNCDELETE AEXIN keyword 389</td>
<td></td>
</tr>
<tr>
<td>SYNCDELETE AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>SYNLST AEXIN keyword 389</td>
<td></td>
</tr>
<tr>
<td>synonym object type 39, 184</td>
<td></td>
</tr>
<tr>
<td>syntax statement conventions 21</td>
<td></td>
</tr>
<tr>
<td>SYSCOPY data set, setting JCL options 115</td>
<td></td>
</tr>
<tr>
<td>SYSDISC data set 115</td>
<td></td>
</tr>
<tr>
<td>SYSERR data set 115</td>
<td></td>
</tr>
<tr>
<td>SYSEXEC AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>SYSIN in BATCH SEARCH 194</td>
<td></td>
</tr>
<tr>
<td>SYSMAP data set 115</td>
<td></td>
</tr>
<tr>
<td>SYSMLIB symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>SYSPROC.ADMIN_DS_LIST stored procedure 168</td>
<td></td>
</tr>
<tr>
<td>SYSPPUNCH data set 115</td>
<td></td>
</tr>
<tr>
<td>SYSREC data sets</td>
<td></td>
</tr>
<tr>
<td>LOB 115</td>
<td></td>
</tr>
<tr>
<td>ROWID 115</td>
<td></td>
</tr>
<tr>
<td>setting JCL options 115</td>
<td></td>
</tr>
<tr>
<td>SYSTEM command 458</td>
<td></td>
</tr>
<tr>
<td>system privilege user object type 39</td>
<td></td>
</tr>
<tr>
<td>system user object type 184</td>
<td></td>
</tr>
<tr>
<td>SYSTEM_MLIB AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>SYSUID symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>SYSUT data set 115</td>
<td></td>
</tr>
<tr>
<td>SYSUT_DATACLASS AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>SYSUT_DATACLASS_ALT AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>SYSUT_EXPDT AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>SYSUT_MGMTCLASS AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>SYSUT_MGMTCLASS_ALT AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>SYSUT_PREFIX AJXPOFIN keyword 433</td>
<td></td>
</tr>
<tr>
<td>SYSUT_PRIQTY AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>SYSUT_RETPD AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>SYSUT_SECQTY AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>SYSUT_STORCLASS AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>SYSUT_STORCLASS_ALT AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>SYSUT_THRESH AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>SYSUT_UNIT AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>SYSUT_UNIT_ALT AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>SYSUTnnn data set 109</td>
<td></td>
</tr>
<tr>
<td>SZDEVT AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>TA command 335</td>
<td></td>
</tr>
<tr>
<td>table constraint object type 41, 184</td>
<td></td>
</tr>
<tr>
<td>table object type 39, 184</td>
<td></td>
</tr>
<tr>
<td>table space mixed object type 41</td>
<td></td>
</tr>
<tr>
<td>table space object type 39, 184</td>
<td></td>
</tr>
<tr>
<td>table space partition object type 41, 44, 184</td>
<td></td>
</tr>
<tr>
<td>table space set object type 41</td>
<td></td>
</tr>
<tr>
<td>table space statistics object type 41</td>
<td></td>
</tr>
<tr>
<td>table spaces</td>
<td></td>
</tr>
<tr>
<td>dropping 272, 274</td>
<td></td>
</tr>
<tr>
<td>recovering structure and data 279</td>
<td></td>
</tr>
<tr>
<td>tables</td>
<td></td>
</tr>
<tr>
<td>CATALOG MANAGER, list of 156</td>
<td></td>
</tr>
<tr>
<td>creating and editing constraints 260</td>
<td></td>
</tr>
<tr>
<td>creating with an existing object 254</td>
<td></td>
</tr>
<tr>
<td>defining columns 256</td>
<td></td>
</tr>
<tr>
<td>for utility support and job generation, list of 156</td>
<td></td>
</tr>
<tr>
<td>identity columns 259</td>
<td></td>
</tr>
<tr>
<td>TABSTATS command 465</td>
<td></td>
</tr>
<tr>
<td>TAILOR command 335, 458</td>
<td></td>
</tr>
<tr>
<td>TAPE_EXPDT AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>TAPE_RETPD AJXPOFIN keyword 434</td>
<td></td>
</tr>
<tr>
<td>TAPE_VOLCNT AJXPOFIN keyword 435</td>
<td></td>
</tr>
<tr>
<td>TAPE1 AJXPOFIN keyword 435</td>
<td></td>
</tr>
<tr>
<td>TAPE2 AJXPOFIN keyword 435</td>
<td></td>
</tr>
<tr>
<td>TAPE3 AJXPOFIN keyword 435</td>
<td></td>
</tr>
<tr>
<td>tapes</td>
<td></td>
</tr>
<tr>
<td>setting default processing options 109</td>
<td></td>
</tr>
<tr>
<td>stacking, disabling 111</td>
<td></td>
</tr>
<tr>
<td>using 118</td>
<td></td>
</tr>
<tr>
<td>TASKID AEXIN keyword 389</td>
<td></td>
</tr>
<tr>
<td>TB command 468</td>
<td></td>
</tr>
<tr>
<td>TB object type 39, 179, 181, 184</td>
<td></td>
</tr>
<tr>
<td>TBBR command 224</td>
<td></td>
</tr>
<tr>
<td>TBCR symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TBCRE symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TBEDIT command 231</td>
<td></td>
</tr>
<tr>
<td>TBNAM symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TBNAME symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TBNODE symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TBF command 468</td>
<td></td>
</tr>
<tr>
<td>TC command 468</td>
<td></td>
</tr>
<tr>
<td>TC object type 41, 184</td>
<td></td>
</tr>
<tr>
<td>TDSN option for site profiles 289</td>
<td></td>
</tr>
<tr>
<td>technical support 3</td>
<td></td>
</tr>
<tr>
<td>TEMPLATE control statements</td>
<td></td>
</tr>
<tr>
<td>available IBM utilities 301</td>
<td></td>
</tr>
<tr>
<td>creating 300</td>
<td></td>
</tr>
<tr>
<td>including in utility jobs 301</td>
<td></td>
</tr>
<tr>
<td>TEMPLATE data set 129</td>
<td></td>
</tr>
<tr>
<td>TEMPLATE descriptor variables, list of 442</td>
<td></td>
</tr>
<tr>
<td>TEMPLATE_DSN AJXPOFIN keyword 435</td>
<td></td>
</tr>
<tr>
<td>temporary work data sets 112</td>
<td></td>
</tr>
<tr>
<td>TEMPUNIT AJXPOFIN keyword 435</td>
<td></td>
</tr>
<tr>
<td>TERM command 463</td>
<td></td>
</tr>
<tr>
<td>THAW command 473</td>
<td></td>
</tr>
<tr>
<td>threshold, for alternate unit 119</td>
<td></td>
</tr>
<tr>
<td>TI symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TI TEMPLATE descriptor variable 450</td>
<td></td>
</tr>
<tr>
<td>TIME OUTPUT descriptor variable</td>
<td></td>
</tr>
<tr>
<td>HMS symbolic variable 443</td>
<td></td>
</tr>
<tr>
<td>JHMS symbolic variable 444</td>
<td></td>
</tr>
<tr>
<td>TI symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TIME symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TIME symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TIME TEMPLATE descriptor variable</td>
<td></td>
</tr>
<tr>
<td>HMS symbolic variable 443</td>
<td></td>
</tr>
<tr>
<td>JHMS symbolic variable 444</td>
<td></td>
</tr>
<tr>
<td>TI symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TIME symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TIME4 symbolic variable 450</td>
<td></td>
</tr>
<tr>
<td>TIMEPARAM AJXPOFIN keyword 435</td>
<td></td>
</tr>
<tr>
<td>TIMESTAMP command 458</td>
<td></td>
</tr>
<tr>
<td>TM command 469</td>
<td></td>
</tr>
<tr>
<td>TM object type 41</td>
<td></td>
</tr>
<tr>
<td>TP command 469</td>
<td></td>
</tr>
<tr>
<td>TP object type 41, 44, 184</td>
<td></td>
</tr>
<tr>
<td>TR command 469</td>
<td></td>
</tr>
<tr>
<td>TR object type 41, 184</td>
<td></td>
</tr>
<tr>
<td>trigger object type 41, 184</td>
<td></td>
</tr>
<tr>
<td>troubleshooting</td>
<td></td>
</tr>
<tr>
<td>attaching and connecting to SSIDs 176</td>
<td></td>
</tr>
<tr>
<td>authorization to perform SEARCH 183</td>
<td></td>
</tr>
<tr>
<td>authorization to use logs 349</td>
<td></td>
</tr>
<tr>
<td>availability of actions and object types 343</td>
<td></td>
</tr>
<tr>
<td>customizing Primary Menu 336</td>
<td></td>
</tr>
<tr>
<td>DB2-identifiers 175</td>
<td></td>
</tr>
<tr>
<td>drop and drop recovery 274</td>
<td></td>
</tr>
<tr>
<td>DROP IS switch 278</td>
<td></td>
</tr>
<tr>
<td>DROP RECOVERY function 279</td>
<td></td>
</tr>
<tr>
<td>Drop Recovery Logs 355</td>
<td></td>
</tr>
<tr>
<td>dropping tables 273</td>
<td></td>
</tr>
<tr>
<td>DSN1COPY to recover data 282</td>
<td></td>
</tr>
<tr>
<td>generating a list 347</td>
<td></td>
</tr>
<tr>
<td>initial list filters 347</td>
<td></td>
</tr>
<tr>
<td>objects excluded from recovery 281</td>
<td></td>
</tr>
<tr>
<td>purging Session Logs 354</td>
<td></td>
</tr>
<tr>
<td>recovering changes from logrba 283</td>
<td></td>
</tr>
<tr>
<td>recovering incremental image copies 283</td>
<td></td>
</tr>
</tbody>
</table>
troubleshooting (continued)
retrieving indexes 273
referential constraints 273
storage space for dropped tables 273
UNKNOWN COMMAND message 340
WHERE clauses in SEARCH 187, 195
WHERE statements with Session Log Lists 352, 356
wildcard characters in qualifiers 42
TRTCH AJXPOFIN keyword 435
TRUNCATE command 461
truncation, long names 84
trusted context attribute object type 40
trusted context authorization ID object type 40
trusted context object type 39
TS command 469
TS object type 39, 179, 184
TS OUTPUT descriptor variable
   IS symbolic variable 444
   IX symbolic variable 444
   IXNAME symbolic variable 444
   IXSPC symbolic variable 444
   SN symbolic variable 449
   SPNAME symbolic variable 449
   TBNAM symbolic variable 450
   TS symbolic variable 450
   TSIX symbolic variable 450
   TSNAM symbolic variable 450
TS symbolic variable 450
TS TEMPLATE descriptor variable
   IS symbolic variable 444
   TBNAM symbolic variable 450
   TBNAME symbolic variable 450
   TS symbolic variable 450
   TSNAM symbolic variable 450
TSCR symbolic variable 450
TSIX symbolic variable 450
TSO ID (exctsoid) 153
TSOPROGRAM AJXPOFIN keyword 435
TSOSUBEXIT AJXPOFIN keyword 435
TSS command 469
TT object type 41
TU1 symbolic variable 450
TU2 symbolic variable 451
TU3 symbolic variable 451
TYPE OUTPUT descriptor variable
   JOBTYP symbolic variable 445
   LOCREM symbolic variable 446
   LR symbolic variable 446
   OBJT symbolic variable 447
   OBJTYP symbolic variable 447
   PB symbolic variable 448
   PRIBAC symbolic variable 448
   RTYPE symbolic variable 448
RUNTYP symbolic variable 449
   TYPE symbolic variable 451
   TYPE parameter 154
   TYPE symbolic variable 451
   TYPES command 469
U
U line command 474
UA command 469
UA object type 41
UCMD symbolic variable 451
UCMD installation option 144
UDOPT symbolic variable 451
UID symbolic variable 451
ULLQ AJXPOFIN keyword 436
ULLQ symbolic variable 451
UN command 469
UN object type 41, 185
UNDO command 473
unit name, ROWID SYSREC data set 118
UNKNOWN COMMAND message 340
UNLD_FREF_DATACLASS AJXPOFIN keyword 436
UNLD_FREF_DIRBLOCK AJXPOFIN keyword 436
UNLD_FREF_MGMTCLASS AJXPOFIN keyword 436
UNLD_FREF_PREFIX AJXPOFIN keyword 436
UNLD_PRIQTY AJXPOFIN keyword 436
UNLD_SECQTY AJXPOFIN keyword 436
UNLD_STORCLASS AJXPOFIN keyword 436
UNLD1_DATACLASS AJXPOFIN keyword 436
UNLD1_DATACLASS_ALT AJXPOFIN keyword 436
UNLD1_EXPDT AJXPOFIN keyword 437
UNLD1_MGMTCLASS AJXPOFIN keyword 437
UNLD1_MGMTCLASS_ALT AJXPOFIN keyword 437
UNLD1_PFX AJXPOFIN keyword 437
UNLD1_PRIQTY AJXPOFIN keyword 437
UNLD1_RETPD AJXPOFIN keyword 437
UNLD1_SECQTY AJXPOFIN keyword 437
UNLD1_STACK AJXPOFIN keyword 437
UNLD1_STORCLASS AJXPOFIN keyword 437
UNLD1_STORCLASS_ALT AJXPOFIN keyword 437
UNLD1_THRESH AJXPOFIN keyword 437
UNLD1_UNIT AJXPOFIN keyword 437
UNLD1_UNIT_ALT AJXPOFIN keyword 438
UNLD2_DATACLASS AJXPOFIN keyword 438
UNLD2_DATACLASS_ALT AJXPOFIN keyword 438
UNLD2_EXPDT AJXPOFIN keyword 438
UNLD2_MGMTCLASS AJXPOFIN keyword 438
UNLD2_MGMTCLASS_ALT AJXPOFIN keyword 438
UNLD2_PFX AJXPOFIN keyword 438
UNLD2_PRIQTY AJXPOFIN keyword 438
UNLD2_RETPD AJXPOFIN keyword 438
UNLD2_SECQTY AJXPOFIN keyword 438
UNLD2_STACK AJXPOFIN keyword 438
UNLD2_STORCLASS AJXPOFIN keyword 438
UNLD2_STORCLASS_ALT AJXPOFIN keyword 438
UNLD2_THRESH AJXPOFIN keyword 439
UNLD2_UNIT AJXPOFIN keyword 439
UNLD2_UNIT_ALT AJXPOFIN keyword 439
UNLD3_DATACLASS AJXPOFIN keyword 439
UNLD3_EXPDT AJXPOFIN keyword 439
UNLD3_MGMTCLASS AJXPOFIN keyword 439
UNLD3_PREFIX AJXPOFIN keyword 439
UNLD3_RETPD AJXPOFIN keyword 439
UNLD3_STORCLASS AJXPOFIN keyword 439
UNLD3_UNIT AJXPOFIN keyword 439
UNLD4_DATACLASS AJXPOFIN keyword 440
UNLD4_MGMTCLASS AJXPOFIN keyword 440
UNLD4_PREFIX AJXPOFIN keyword 440
UNLD4_STORCLASS AJXPOFIN keyword 440
UNLD4_UNIT AJXPOFIN keyword 440
UNLOAD command 288, 463
unload data set, used by utilities 108
UNLOAD+_LOAD AJXPOFIN keyword 403, 440
UNLOADDOPT AEXIN keyword 389
UNLOADDOPT AJXPOFIN keyword 403, 440
UP command 473
UP ISPF command 46
UPART symbolic variable 451
UPDATE command 461
US command 469
US object type 39, 185
user authorization object type 41
user command program 153
user commands table
  modifying 148
  retaining from previous release 151
user object type 39, 185
user options
  general 72
  JCL Generation 73
  setting 76
user POF
  creating 132
  updating directly 133
  updating in options panels 133
  using 134
user privileges, copying by privilege type 317
user profile data set for user profiles 289
user variables 130
USER_VAR1_CHAR AJXPOFIN keyword 440
USER_VAR2_CHAR AJXPOFIN keyword 440
USER_VAR3_CHAR AJXPOFIN keyword 440
USER_VAR4_CHAR AJXPOFIN keyword 440
USER_VAR5_CHAR AJXPOFIN keyword 440
USERID symbolic variable 451
USER2 symbolic variable 451
USERID OUTPUT descriptor variable
  USERID symbolic variable 451
  ZACCTNUM symbolic variable 453
ZPREFIX symbolic variable 453
USERID symbolic variable 451
user-written commands
  &CLIST parameter 145
  &CMD parameter 146
  &DB2MAX parameter 146
  &DB2MIN parameter 146
  &HELP parameter 146
  &LOAD parameter 146
  &LOG parameter 146
  &LSTO parameter 146
  &NLIST parameter 146
  &NOSERVER parameter 147
  &NUM parameter 147
  &OBJECTS parameter 147
  &PARSE parameter 147
  &PLAN parameter 147
  &WFEK parameter 148
commands table 143
  creating 148
  object types 155
  writing as CLIST 151
  writing as program 150
USRCOMND member 148, 152
UT symbolic variable 452
UT TEMPLATE descriptor variable
  ALID symbolic variable 442
  FCMD symbolic variable 443
  JQID symbolic variable 446
  UCMDF symbolic variable 451
  USER1 symbolic variable 451
  USER2 symbolic variable 451
  UTID symbolic variable 452
  UTIL symbolic variable 452
  UTILID symbolic variable 452
  UTILPFX symbolic variable 452
  UTILSFX symbolic variable 452
  WKID symbolic variable 452
  WORKID symbolic variable 452
  WORKID8 symbolic variable 453
  UTID OUTPUT descriptor variable 452
  UTID symbolic variable 452
  UTIL command 464
UTIL OUTPUT descriptor variable
  ALID symbolic variable 442
  FCMD symbolic variable 443
  JQID symbolic variable 446
  LI symbolic variable 446
  LIST symbolic variable 446
  OBJT symbolic variable 447
  OBJTYP symbolic variable 447
  UCMDF symbolic variable 451
  USER1 symbolic variable 451
  USER2 symbolic variable 451
  UT symbolic variable 452
UTID symbolic variable 452
UTIL symbolic variable 452
UTILID symbolic variable 452
UTILPFX symbolic variable 452
UTILSFX symbolic variable 452
WKID symbolic variable 452
WORKID symbolic variable 452
WORKID8 symbolic variable 453
UTIL profile ID command 464
UTIL symbolic variable 452
UTILITY command 294, 295, 464
utility commands, list of 461
utility edit propagation 298
utility module names, setting default processing options 124
utility processing
  BMC Software utilities 286
  IBM DB2 utilities 287
  JCL data set name 290
  JCL options 291
  multiple utilities 294
  number of control statements allowed 288
  single utility 289
  User Profile data set name 290
  Utility ID variables 291, 292
UTILITY profile ID command 464
utility profiles
  allocating a data set 289
  changing options values 298
  creating from an existing profile 296
  editing 298
  last-used profile ID 299
  online tutorial 288
  PROFILE command 300
  profile ID 293
  setting up 289
  site profiles 288
  User Profile data set name 290
  user profiles 288
UTILPFX symbolic variable 452
UTILSFX symbolic variable 452
UVR1 symbolic variable 452
UVR2 symbolic variable 452
UVR3 symbolic variable 452
UVR4 symbolic variable 452
UVR5 symbolic variable 452
virtual storage 71
VL command 469
VL object type 41, 185
volume object type 41, 185
VSAM object
  data set sizing 107
  JCL generation 363
  sampling 363
VW command 469
VW object type 39, 179, 181, 185

W
Wait-for-Enter commands 59–63
WARNRC AEXIN keyword 389
WHERE clauses in SEARCH 187
where command is executed parameter (exccsrc) 153
wildcard characters
  cascading authorizations 309
  fixed-length CHAR columns 43
  in host variables 191
  in quick-search 194
  in WHERE clause 187
  two-part object names 44
  use in Qualifier field 42
  with saved SEARCH variables 193
WKID symbolic variable 452
WKOWN symbolic variable 452
WKOWNER symbolic variable 452
work data sets
  permanent 115
  temporary 112
  used by utilities 108
WORK_DATACLASS AJXPOFIN keyword 441
WORK_MGMTCCLASS AJXPOFIN keyword 441
WORK_STORCLASS AJXPOFIN keyword 441
WORKID AEXIN keyword 389
WORKID symbolic variable 452
WORKID8 symbolic variable 453
worklists, executing 70
WRKnnn data set 109

X
X ALL command 182
X command 458
X line command 182
X line designator 62
XML relationship object type 41
XML string object type 41
XR command 469
XR object type 41
XS command 469
XS object type 41
XT command 469  
XT object type 39, 183

Y
YE symbolic variable 453
YE TEMPLATE descriptor variable 453
YEAR OUTPUT descriptor variable 453
YEAR symbolic variable 453
YMD symbolic variable 453
YY OUTPUT descriptor variable 453
YY symbolic variable 453
YYDDD symbolic variable 453
YYYYDDD symbolic variable 453

Z
Z line command 474
ZACCTNUM SLIB variable 453
ZACCTNUM symbolic variable 453
ZOOM command 253, 407, 473
ZOOM ISPF command 46
ZPREFIX SLIB variable
   PREFIX symbolic variable 448
   ZPREFIX symbolic variable 453
ZPREFIX symbolic variable 453
ZSYSID SLIB variable 454
ZSYSID symbolic variable 454
ZUSER SLIB variable
   UID symbolic variable 451
   USERID symbolic variable 451
   ZUSER symbolic variable 454
   ZUSER symbolic variable 454