CATALOG MANAGER for DB2®
User Guide

Supporting

Version 11.1 Administrative Assistant for DB2
Version 11.1 CATALOG MANAGER for DB2
Version 10.2 Database Administration for DB2
Version 11.1 System Performance for DB2

June 2013
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**

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>1 713 918 8800 or 1 800 841 2031</td>
<td>1 713 918 8000</td>
</tr>
<tr>
<td>2101 CITYWEST BLVD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Outside United States and Canada**

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>+01 713 918 8800</td>
<td>+01 713 918 8000</td>
</tr>
</tbody>
</table>

© Copyright 1998-2013 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 UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES.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 Support Central 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 1 800 537 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 ............................................................. 19
Conventions ........................................................................... 20
Syntax statements ................................................................. 20

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 ............................. 38
Selecting an Action ............................................................ 38
Identifying an object type ................................................... 38
Specifying a qualifier ........................................................ 42
Using nonprintable or nonviewable characters ...................... 44
Setting the JCL options for tapes ............................................... 109
Setting the JCL options for temporary work data sets .................. 111
Setting the JCL options for permanent data sets ....................... 114
Setting the JCL generation data group options ......................... 119
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 ............... 130
Setting user variables ....................................................... 131
Creating a user POF .......................................................... 133
Updating a user POF .......................................................... 134
Using multiple POFs .......................................................... 135
Refreshing the initial POF ................................................... 136
Generating POF reports ..................................................... 137
Reusing a POF in a subsequent installation ............................. 138
Overriding POF values in SLIBs ........................................... 139
Adding steps to the JCL ....................................................... 141
Obtaining a list of TEMPLATEs or LISTDEFs in CATALOG MANAGER . 144
Using the commands table .................................................. 144
Setting up and modifying the user commands table ................ 146
Commands table syntax and parameters ................................ 149
Migrating a user commands table to a new release of CATALOG MANAGER ................................................. 152
Writing user commands as CLISTs ........................................ 153
Development aids for user commands .................................... 153
Command program parameters ............................................ 154
Passing object type and name ............................................. 155
CATALOG MANAGER tables ............................................... 157
Where to go from here ....................................................... 158

Chapter 4 Accessing other DB2 subsystems ................................. 159
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
Chapter 7 Creating objects 251
Overview ................................................................. 251
Before you begin .................................................. 251
Estimating space requirements based on user-specified values ........................................... 252
  Estimating space requirements for a table space ......................................................... 253
  Estimating space requirements for an index .............................................................. 256
Using an existing object as a model to create objects ...................................................... 257
  Considerations for creating objects .............................................................. 257
  Example of creating a table .............................................................................. 258
Generating DDL to create objects ........................................................................ 271
Where to go from here ................................................. 274

Chapter 8 Dropping and recovering objects 275
Overview ................................................................. 275
Considerations for dropping objects ........................................................................ 275
  Considerations for dropping a table space .............................................................. 276
  Considerations for dropping pending changes ....................................................... 276
  Considerations for dropping a table ...................................................................... 276
  Considerations for dropping an index ................................................................... 277
Simulating a drop ................................................................................................. 278
Dropping an object ............................................................................................... 278
Recovering an object and its data ............................................................................... 283
Where to go from here ................................................. 288

Chapter 9 Generating utility jobs 289
Overview ................................................................. 289
Available utilities ............................................................................................... 290
  BMC products .............................................................................................. 290
  IBM DB2 utilities ....................................................................................... 291
Issuing utility commands ....................................................................................... 292
Utility profiles ..................................................................................................... 292
  Creating a user utility profile data set ...................................................................... 293
  Creating a utility profile for a single utility ......................................................... 295
  Creating a utility profile for multiple utilities ..................................................... 300
  Creating a utility profile from an existing profile ............................................... 302
Editing a utility profile ......................................................................................... 304
Working with the last-used utility profile .............................................................. 305
Switching utility profiles ....................................................................................... 306
Chapter 10  Issuing DB2 commands  311
Overview ............................................................... 311
Methods for issuing commands .................................. 311
  Using command prompts ........................................... 312
  Using model commands ............................................ 315
  Specifying command syntax ..................................... 316
DB2 command profiles ............................................. 318
Where to go from here .............................................. 320

Chapter 11  Managing authorizations  321
Overview ............................................................... 321
Granting privileges by using the GRANT commands ........... 322
  Granting privileges on a table .................................... 323
  Granting privileges on a hierarchy of DB2 objects .......... 327
  Importing the SQL in another subsystem ...................... 330
  Copying an SQL_Table entry to another subsystem .......... 332
Granting privileges by issuing the COPYAUTHS command ...... 333
  Copying user ID privileges ....................................... 333
  Copying object privileges ....................................... 337
Granting privileges by generating SQL .......................... 337
Revoking privileges ................................................ 338
  Preserving access to information ............................... 339
  Generating the cascade report .................................. 339
  Revoking privileges on specific objects ....................... 342
  Verifying current authorizations ............................... 348
Where to go from here .............................................. 349

Chapter 12  Customizing CATALOG MANAGER command access  351
Overview ............................................................... 351
Preparing to implement session profiles ....................... 352
Creating session profiles ......................................... 352
  Commands to create and edit session profiles ................ 353
  Creating a session profile with a customized primary menu .. 354
  Displaying the session profiles list ........................... 356
  Displaying session profile descriptions ....................... 357
  Creating a session profile with a tailored commands table .. 358
  Creating a session profile with an initial list filter .......... 359
Editing session profiles .......................................... 360
  Deleting session profiles ........................................ 360
  Editing a customized Primary Menu ............................ 361
  Editing a tailored commands table ............................ 362
  Retrieving an initial list filter ................................ 363
## Assigning session profiles to users

Assigning session profiles to users........................................... 363

## Determining a user’s capabilities

Determining a user’s capabilities............................................. 363

## Activating and deactivating session profiles

Activating and deactivating session profiles............................... 365

## Where to go from here

Where to go from here............................................................ 366

### Chapter 13 Maintaining logs

Overview......................................................................................... 367

Accessing the logs........................................................................ 368

Using the Session Log

- Browsing the Session Log.......................................................... 369
- Purging the Session Log............................................................ 372

Using the DDL Audit Log.............................................................. 374

Using the Drop Recovery Log....................................................... 374

- Browsing the Drop Recovery Log............................................... 375
- Purging the Drop Recovery Log.................................................. 378

Where to go from here............................................................... 378

### Appendix A Generating JCL

The JCL Generation function....................................................... 379

- Modifying the ISPF skeletons and variables.............................. 380
- Sizing JCL Generation data sets................................................. 381

### Appendix B Using the Skeleton Library compiler

Overview......................................................................................... 383

- Compiling SLIBs........................................................................ 384
- Changing SLIBs......................................................................... 385
  - Testing changes using ISPF file tailoring............................... 386
  - Compiling changed SLIBs....................................................... 387

- Processing SLIBs....................................................................... 387
  - Generating the SLIB report.................................................... 387

### Appendix C Integrating CATALOG MANAGER for DB2 with the Common Explain component

Overview......................................................................................... 391

Before you begin........................................................................... 392

Commands to access SQL............................................................. 392

- Explaining a DBRM, package, or plan....................................... 394
- Explaining a statement from a DBRM or package....................... 395
- Explaining a statement from the SQL_Table................................ 398

- Editing the SQL statement by using the SQLX Edit Macro........... 400

### Appendix D JCL Generation keywords and variables

AEXIN keywords........................................................................... 403

Symbolic variables....................................................................... 409
Appendix E  CATALOG MANAGER installation options  423
Overview ................................................................. 423
Installation options ............................................... 423

Appendix F  JCL Generation product options  439
Overview ................................................................. 439
Product options ................................................... 439

Appendix G  CATALOG MANAGER worklist commands  523
Worklist overview ................................................... 523
Worklist file format .................................................. 524
Worklist commands .................................................. 525
-BMCU—Run a BMC utility ...................................... 525
-DBUG—Debug ......................................................... 526
-DSN1—DSN1COPY (IBM utility) .............................. 526
-MERG—MERGECOPY (IBM utility) ......................... 527
-MODI—MODIFY (IBM utility) ................................ 528
-NOOP—No operation ............................................. 528
-QUI—QUIESCE (IBM utility) ................................. 529
-REPO—REPORT (IBM utility) ............................... 530
-REPX—REPAIR (IBM utility) ............................... 530
-SQLM (Group multiple ALTER statements) ............. 531
-STOS—STOSPACE (IBM utility) .......................... 531

Appendix H  Commands  533
CATALOG MANAGER commands ............................... 533
DB2 action commands .............................................. 537
Utility commands ..................................................... 540
Utility list commands ............................................. 543
Statistics commands ............................................... 544
List commands ....................................................... 544
User commands ....................................................... 549
Data browsing and editing commands ....................... 549
  Command-line commands .................................. 549
  Line commands .................................................. 552

Glossary  555

Index  563
Figures

Example of BMC Software Administrative Products for DB2 panel .................. 35
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
DB2 Special Registers panel ............................................................... 68
How CATALOG MANAGER uses options ............................................. 74
Options Dataset not defined panel ......................................................... 77
Allocate Data Set .................................................................... 78
CATALOG MANAGER Options panel ..................................................... 79
Options Dataset has changed panel ....................................................... 80
General Options panel ................................................................. 82
Object Use Options panel ................................................................ 84
JCL Generation Options panel ............................................................ 86
Datasets panel .................................................................... 88
SQL and Confirm Options panel ........................................................... 90
SQL Select panel .................................................................... 92
Colors panel .................................................................... 94
Switches panel .................................................................... 95
Describe Options panel .................................................................. 98
JCL Generation Update Panel .............................................................. 99
JCL Generation Jobcard Options Update panel ...................................... 101
JCL Generation STEPLIB Options Update panel .................................. 103
JCL Generation Static Data Set Options panel ...................................... 105
JCL Generation Tape Options Update panel ........................................ 110
JCL Generation Options For Sort Files Update panel ............................... 112
JCL Generation Data Set Options For Sortout Update panel ....................... 115
Using the &GDG symbolic variable ........................................ 116
Data set names resolved with the &GDG symbolic variable ................. 117
JCL Generation Generation Data Group Options Update panel .............. 120
JCL Generation Debugging, Display And Execution Options Update panel .... 122
Execution JCL with comments ............................................. 123
JCL Generation BMC Utility Option Module Names Update panel ........... 125
Online Reorg Utility Options panel ........................................ 126
PROC and STEP Names panel ............................................. 128
JCL Generation IBM Utility Dynamic Data Set Options Update panel ....... 131
User Defined Variables panel ............................................. 132
JCL Generation Product Options File (POF) Functions panel ............... 133
Reusing an existing POF .................................................. 138
Specifying the name of the existing POF .................................. 139
Specifying the prefix for a copy data set .................................. 140
Changing the SLIB variable for the copy data set in AJX#DSNS ............. 140
JCL Generation Jobcard Options Update panel ................................ 142
Including the JCL library .................................................. 142
JCL Generation Debugging, Display and Execution Options Update panel .... 143
Including JCL between steps ............................................... 143
Including JCL at the end of the jobs ...................................... 144
Expanded INCLUDE members .............................................. 144
Sample ACTCOMND file .................................................... 145
STATUS command ......................................................... 147
DISPLAY UTILITY command ............................................... 147
New STATUS command .................................................... 147
Attaching to another DB2 subsystem ...................................... 160
Change Access panel showing DB2 attach specifications ................. 161
Attaching and connecting to other DB2 subsystems .......................... 165
Change Access panel after new connection .................................. 167
Connections List panel ..................................................... 171
Section of Connections Table panel ...................................... 173
Table Space List panel to generate a mixed list ............................. 180
Mixed Object List panel generated from table space list ................... 181
Combined package list ..................................................... 183
Search panel for table spaces ............................................ 187
Search Options panel - saving search variables ............................ 189
Search Options panel - retrieving search variables ......................... 190
SQL Host Variables List panel ............................................ 193
Column Order Specification panel ....................................... 199
Sort Specifications panel .................................................. 201
Table Count List panel ..................................................... 203
DB2 Catalog Counts panel .................................................. 204
Describe Table panel generated from DESCRIBE command ............... 206
Describe Table panel generated from DESTATISTICS command ............ 208
CATALOG MANAGER Batch Job panel .................................... 211
NO CATALOG MANAGER COMMANDS message in JCL .................... 214
Confirm SQL panel for CREATE table .................................... 216
SQL_Table List panel ....................................................... 217
Confirm APPLY SQL MODEL panel ....................................... 220
<table>
<thead>
<tr>
<th>Panel Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confirm SQL panel for authorization processing</td>
<td>221</td>
</tr>
<tr>
<td>Confirm SQL panel for extended SQL processing</td>
<td>223</td>
</tr>
<tr>
<td>SQL Host Variables List panel</td>
<td>224</td>
</tr>
<tr>
<td>Browse DB2 Table Options panel</td>
<td>229</td>
</tr>
<tr>
<td>Select Statement Specification panel</td>
<td>230</td>
</tr>
<tr>
<td>Edit and Browse Options panel</td>
<td>232</td>
</tr>
<tr>
<td>Edit DB2 Table Options panel</td>
<td>236</td>
</tr>
<tr>
<td>Edit DB2 Table panel in column view mode</td>
<td>239</td>
</tr>
<tr>
<td>Display More Rows Options panel</td>
<td>240</td>
</tr>
<tr>
<td>COPYRIGHT From DB2 Table Options panel</td>
<td>242</td>
</tr>
<tr>
<td>Select Statement Specification panel</td>
<td>243</td>
</tr>
<tr>
<td>Edit DB2 Table panel showing rows that were copied</td>
<td>244</td>
</tr>
<tr>
<td>Copy Table Rows Specifications panel</td>
<td>246</td>
</tr>
<tr>
<td>Select Statement Specification panel</td>
<td>247</td>
</tr>
<tr>
<td>ISPF panel for editing INSERT statement</td>
<td>248</td>
</tr>
<tr>
<td>Tablespace Estimation panel</td>
<td>253</td>
</tr>
<tr>
<td>Create/Alter Table panel</td>
<td>259</td>
</tr>
<tr>
<td>Create/Alter Table panel</td>
<td>260</td>
</tr>
<tr>
<td>Columns panel</td>
<td>261</td>
</tr>
<tr>
<td>Column Options panel</td>
<td>262</td>
</tr>
<tr>
<td>Additional column options panel</td>
<td>263</td>
</tr>
<tr>
<td>Column Identity Information panel</td>
<td>264</td>
</tr>
<tr>
<td>Table Constraints panel</td>
<td>264</td>
</tr>
<tr>
<td>Materialized Query Options panel</td>
<td>266</td>
</tr>
<tr>
<td>Select Generate Text panel</td>
<td>267</td>
</tr>
<tr>
<td>Confirm SQL panel for creating a table</td>
<td>269</td>
</tr>
<tr>
<td>Confirm DROP panel for DROP TABLE SPACE procedure</td>
<td>279</td>
</tr>
<tr>
<td>Drop Dependency List</td>
<td>280</td>
</tr>
<tr>
<td>Tablespace List panel after drop</td>
<td>282</td>
</tr>
<tr>
<td>Drop Recovery List panel for dropped table spaces</td>
<td>284</td>
</tr>
<tr>
<td>Recovery Statements panel</td>
<td>285</td>
</tr>
<tr>
<td>Describe Audit Log Entry panel for the Drop Recovery Log</td>
<td>286</td>
</tr>
<tr>
<td>Utility List panel</td>
<td>296</td>
</tr>
<tr>
<td>REORG PLUS utility panel</td>
<td>298</td>
</tr>
<tr>
<td>Utility List panel after editing utility statements</td>
<td>299</td>
</tr>
<tr>
<td>Utility Selections panel</td>
<td>301</td>
</tr>
<tr>
<td>Profile selection panel</td>
<td>303</td>
</tr>
<tr>
<td>Listdef/Template options panel</td>
<td>308</td>
</tr>
<tr>
<td>Listdef selection list panel</td>
<td>309</td>
</tr>
<tr>
<td>DB2 Command Prompts panel</td>
<td>312</td>
</tr>
<tr>
<td>Start Database panel</td>
<td>313</td>
</tr>
<tr>
<td>DB2 Commands panel</td>
<td>314</td>
</tr>
<tr>
<td>DB2 command syntax</td>
<td>316</td>
</tr>
<tr>
<td>PROFILE command syntax</td>
<td>318</td>
</tr>
<tr>
<td>DB2 command profile</td>
<td>318</td>
</tr>
<tr>
<td>Table List panel with GRANT command</td>
<td>324</td>
</tr>
<tr>
<td>Grant Table Privileges panel</td>
<td>324</td>
</tr>
<tr>
<td>Confirm SQL panel for granting table privileges</td>
<td>326</td>
</tr>
<tr>
<td>Confirm SQL panel for authorization processing</td>
<td>329</td>
</tr>
</tbody>
</table>
Import SQL from a PDS panel .................................................. 331
Copy User Authorizations panel ............................................. 334
Confirm SQL panel for copying authorizations .......................... 335
Cascade Report .............................................................. 340
Object privileges panel ...................................................... 342
User List panel ............................................................. 343
Confirm SQL for Revoke Reassign panel ................................ 344
Confirm SQL for Revoke Reassign Grants panel ......................... 346
Cascade List Report for Revoke/Reassign ................................. 347
Menu Profile Customizing panel ........................................... 354
Preview of customized Primary Menu panel .............................. 355
Profiles List panel ........................................................... 356
Describe Profile Entries panel .............................................. 357
Section of Profile Command Tailoring List panel ......................... 359
Menu Profile Customizing panel to edit a session profile .......... 361
Customized Primary Menu panel .......................................... 366
Log Maintenance Menu panel .............................................. 368
Browse Session Log panel .................................................. 369
Session Log List ............................................................. 371
Describe Audit Log Entry panel for a Session Log ..................... 372
Purge Session Log panel ................................................... 373
Browse Recovery Log panel ............................................... 375
Recovery Log List ........................................................... 376
Recovery Log Detail ........................................................ 377
Describe Audit Log Entry panel for a Drop Recovery Log ............ 378
Processing flow of the SLIB compiler .................................... 384
Sample runtime report ..................................................... 388
Using the BMCEXPLORE command ...................................... 392
Using the DESCRIBE and GET commands ............................... 393
Using the ANALYZE command .......................................... 394
BMCEXPLORE command issued against plan ......................... 395
DESCRIBE panel ............................................................ 396
Use of GET subcommand .................................................. 397
Confirm SQL panel ........................................................ 397
SQL_Table List panel ....................................................... 399
Confirm SQL panel ........................................................ 399
ISPF Edit panel ............................................................. 400
CATALOG MANAGER installation options module ..................... 423
Product options file ........................................................ 439
-BMCU command .......................................................... 526
-DSN1 command ........................................................... 527
-MERG command .......................................................... 527
-MODI command—MODIFY RECOVERY ................................. 528
-MODI command—MODIFY STATISTICS ............................... 528
-NOOP command ........................................................... 529
-QUI command ............................................................ 529
-REPO command .......................................................... 530
–SQLM command ............................................................ 531
-STOS command ............................................................ 531
Tables

Description of Primary Menu panel .................................................. 37
Object types displayed on the Primary Menu panel .............................. 39
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 ................................................................................... 75
Valid values for the Default SQLID field ........................................... 81
Valid values for the SET CURRENT SQLID to field ............................... 84
Data set sizing options ........................................................................ 106
Work data sets in the JCL cleanup job step ......................................... 107
Work data sets used by utilities ........................................................... 108
Values for 7-track tape drives ............................................................. 110
POF keywords used to suppress the ddname ........................................ 116
Comment codes for data set sizing .................................................... 123
Syntax of commands table entry ........................................................ 149
Commands table variables ................................................................. 149
Customizable code for building user commands ................................... 153
Key parameters for a user command program ..................................... 154
Parameters for use with the $ACTULOG macro ................................... 155
Passing object types and names in user-written commands ................... 156
ISPF variables for user commands or CLISTs ..................................... 157
CATALOG MANAGER tables ............................................................... 157
CONNECT command parameters ....................................................... 167
Columns on the Connections Table panel .......................................... 174
Valid source objects for mixed lists ................................................... 179
Valid source objects for generating combined lists ............................... 182
Valid source objects for searches ....................................................... 185
Valid search operators ........................................................................ 187
Host variable values ........................................................................... 193
DES command descriptions ............................................................... 207
DB2 command syntax ......................................................................... 213
Commands to invoke data browsing .................................................... 228
SELECT statement specifications ....................................................... 230
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT statement specification panel commands</td>
<td>231</td>
</tr>
<tr>
<td>Additional edit and browse options</td>
<td>233</td>
</tr>
<tr>
<td>Commands to invoke data editing</td>
<td>235</td>
</tr>
<tr>
<td>Data definition language commands</td>
<td>271</td>
</tr>
<tr>
<td>Statements excluded from object recovery</td>
<td>285</td>
</tr>
<tr>
<td>Commands to invoke BMC products</td>
<td>290</td>
</tr>
<tr>
<td>Commands to invoke IBM DB2 utilities</td>
<td>291</td>
</tr>
<tr>
<td>Utility ID variables</td>
<td>297</td>
</tr>
<tr>
<td>Status column values for the Utility List panel</td>
<td>300</td>
</tr>
<tr>
<td>Valid IBM utilities for TEMPLATE and LISTDEF statements</td>
<td>307</td>
</tr>
<tr>
<td>Process type options</td>
<td>313</td>
</tr>
<tr>
<td>Commands used on DB2 Commands panel</td>
<td>316</td>
</tr>
<tr>
<td>DB2 command variables</td>
<td>317</td>
</tr>
<tr>
<td>Source objects for the DCL command</td>
<td>338</td>
</tr>
<tr>
<td>Features supported by session profiles</td>
<td>352</td>
</tr>
<tr>
<td>Session profile commands</td>
<td>353</td>
</tr>
<tr>
<td>User capabilities determined by CRS DOPT and PR parameter</td>
<td>364</td>
</tr>
<tr>
<td>User-defined variables</td>
<td>381</td>
</tr>
<tr>
<td>Data set sizing values and sources</td>
<td>382</td>
</tr>
<tr>
<td>Runtime report statistics</td>
<td>389</td>
</tr>
<tr>
<td>AEXIN keywords</td>
<td>403</td>
</tr>
<tr>
<td>Symbolic variables</td>
<td>409</td>
</tr>
<tr>
<td>$ACTSQLD options</td>
<td>434</td>
</tr>
<tr>
<td>CATALOG MANAGER commands</td>
<td>533</td>
</tr>
<tr>
<td>DB2 action commands</td>
<td>537</td>
</tr>
<tr>
<td>Utility commands</td>
<td>540</td>
</tr>
<tr>
<td>Utility list commands</td>
<td>543</td>
</tr>
<tr>
<td>Statistics commands</td>
<td>544</td>
</tr>
<tr>
<td>List commands</td>
<td>544</td>
</tr>
<tr>
<td>User commands</td>
<td>549</td>
</tr>
<tr>
<td>Command-line commands for browsing or editing data</td>
<td>549</td>
</tr>
<tr>
<td>Line commands for browsing or editing data</td>
<td>552</td>
</tr>
</tbody>
</table>
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 support website at http://www.bmc.com/support.

---

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

From the BMC Support Central website (http://www.bmc.com/support), you can use either of the following methods to access related publications that support your product or solution:

- Link to the BMC Documentation Center (https://webapps.bmc.com/infocenter/index.jsp) to browse documentation sets

- View BMC Quick Course Demos (short overviews of selected product concepts, tasks, or features), which are included in the BMC Documentation Center
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:
  
  ```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><em>alias</em></td>
</tr>
<tr>
<td>a name or value. If a variable is represented by two or more words,</td>
<td><em>databaseDirectory</em></td>
</tr>
<tr>
<td>initial capitals distinguish the second and subsequent words.</td>
<td><em>serverHostName</em></td>
</tr>
<tr>
<td>Brackets indicate a group of optional items. Do not type the</td>
<td><code>[tableName, columnName, field]</code></td>
</tr>
<tr>
<td>brackets when you enter the option. A comma means that you can</td>
<td><code>[-full, -incremental, -level]</code></td>
</tr>
<tr>
<td>choose one or more of the listed options. You must use a comma to</td>
<td>(UNIX)</td>
</tr>
<tr>
<td>separate the options if you choose more than one option.</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Example</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Braces indicate that at least one of the enclosed items is required. Do not type the braces when you enter the item.</td>
<td>(DBDName</td>
</tr>
<tr>
<td></td>
<td>UNLOAD device={disk</td>
</tr>
<tr>
<td></td>
<td>{-a</td>
</tr>
<tr>
<td>A vertical bar means that you can choose only one of the listed items. In the example, you would choose either commit or cancel.</td>
<td>{commit</td>
</tr>
<tr>
<td></td>
<td>{-commit</td>
</tr>
<tr>
<td>An ellipsis indicates that you can repeat the previous item or items 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, Installation System User Guide, and BMC Products and Solutions for DB2 Configuration 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, Installation System User Guide, and BMC Products and Solutions for DB2 Configuration 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, Installation System User Guide, and BMC Products and Solutions for DB2 Configuration 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.
Where to go from here
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 ...................... 38
   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 System User Guide and the BMC Products and Solutions for DB2 Configuration Guide.

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

**NOTE**

For more information, view the Quick Course Getting Started. You must have a BMC Support ID to view the Quick Course.

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.
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.
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).

Press **Enter**.

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

**Figure 2  CATALOG MANAGER 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**.
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 F, “JCL Generation product options,” 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, \texttt{>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.

\textit{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 2**  
**Object types displayed on the Primary Menu panel**

<table>
<thead>
<tr>
<th>Object type code</th>
<th>Object type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>database</td>
</tr>
<tr>
<td>SG</td>
<td>storage group</td>
</tr>
<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&lt;sup&gt;a&lt;/sup&gt;</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&lt;sup&gt;b&lt;/sup&gt;</td>
<td>auxiliary table</td>
</tr>
</tbody>
</table>

<sup>a</sup> The product displays the object type on the panel if the Distributed Data Facility (DDF) is defined to CATALOG MANAGER.

<sup>b</sup> 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&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>BP</td>
<td>buffer pool</td>
</tr>
<tr>
<td>CA</td>
<td>column authorization</td>
</tr>
</tbody>
</table>
### Table 3  Object types not displayed on the Primary Menu panel (Part 2 of 3)

<table>
<thead>
<tr>
<th>Object type code</th>
<th>Object type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD</td>
<td>check dependent</td>
</tr>
<tr>
<td>CL</td>
<td>column label&lt;sup&gt;a&lt;/sup&gt;</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>CXA</td>
<td><em>(DB2 Version 9 or later)</em> trusted context authorization ID&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>CXT</td>
<td><em>(DB2 Version 9 or later)</em> trusted context attribute&lt;sup&gt;b&lt;/sup&gt;</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><em>(DB2 Version 9 or later)</em> dependency</td>
</tr>
<tr>
<td>DS</td>
<td>data set&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>DT</td>
<td>data or distinct type</td>
</tr>
<tr>
<td>EN</td>
<td><em>(DB2 Version 9 or later)</em> environment variables</td>
</tr>
<tr>
<td>FK</td>
<td>foreign key&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>FN</td>
<td>function routine</td>
</tr>
<tr>
<td>FO</td>
<td>routine option&lt;sup&gt;c&lt;/sup&gt;</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 copy&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>IL</td>
<td>IP list</td>
</tr>
<tr>
<td>IM</td>
<td>index mixed&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>IN</td>
<td>IP name&lt;sup&gt;d&lt;/sup&gt;</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 statistics&lt;sup&gt;e&lt;/sup&gt;</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><em>(DB2 Version 9 or later)</em> Java path</td>
</tr>
<tr>
<td>KC</td>
<td>key column&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>KT</td>
<td><em>(DB2 Version 9 or later)</em> key-target</td>
</tr>
<tr>
<td>KTD</td>
<td><em>(DB2 Version 9 or later)</em> key-target distribution&lt;sup&gt;f&lt;/sup&gt;</td>
</tr>
<tr>
<td>KTH</td>
<td><em>(DB2 Version 9 or later)</em> key-target history&lt;sup&gt;f&lt;/sup&gt;</td>
</tr>
<tr>
<td>KTS</td>
<td><em>(DB2 Version 9 or later)</em> key-target statistics&lt;sup&gt;f&lt;/sup&gt;</td>
</tr>
<tr>
<td>KU</td>
<td>key column user</td>
</tr>
<tr>
<td>LK</td>
<td>limit key&lt;sup&gt;ag&lt;/sup&gt;</td>
</tr>
<tr>
<td>LL</td>
<td>LU list&lt;sup&gt;d&lt;/sup&gt;</td>
</tr>
<tr>
<td>LM</td>
<td>LU mode&lt;sup&gt;d&lt;/sup&gt;</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>LS</td>
<td>LU modeselect(^d)</td>
</tr>
<tr>
<td>LU</td>
<td>LU names(^d)</td>
</tr>
<tr>
<td>MQT</td>
<td>materialized query table(^c)</td>
</tr>
<tr>
<td>MX</td>
<td>mixed object types(^a)</td>
</tr>
<tr>
<td>NP</td>
<td>native SQL procedure</td>
</tr>
<tr>
<td>OB</td>
<td>online schema changes(^c)</td>
</tr>
<tr>
<td>OS</td>
<td>LOBSTATS</td>
</tr>
<tr>
<td>PA</td>
<td>plan authorization(^a)</td>
</tr>
<tr>
<td>PI</td>
<td>packlist(^a)</td>
</tr>
<tr>
<td>PK</td>
<td>primary key(^a)</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(^a)</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(^{ah})</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(^a)</td>
</tr>
<tr>
<td>UA</td>
<td>user authorization(^a)</td>
</tr>
<tr>
<td>UN</td>
<td>user name(^d)</td>
</tr>
<tr>
<td>VL</td>
<td>volume</td>
</tr>
<tr>
<td>XC</td>
<td>BMC Software COPY PLUS for DB2 cabinet copy</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.
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.

---

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>

---

h This object code displays only the tables in the table space that have referential integrity.
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 6 Wildcards in fixed-length CHAR columns

<table>
<thead>
<tr>
<th>If the qualifier is</th>
<th>Does it match . . . ABC . . ?</th>
<th>Does it match ABC . . . ?</th>
<th>Does it match . . . . . . . ABC?</th>
</tr>
</thead>
<tbody>
<tr>
<td>%ABC%</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ABC%</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>%ABC</td>
<td>No</td>
<td>No</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 DTBCREATOR.DTBNAME.</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>END (or F3)</td>
<td>validates and processes information, the same as the <strong>Enter</strong> 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 <strong>Enter</strong> 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 10  ISPF commands (Part 2 of 2)

<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.</td>
</tr>
<tr>
<td></td>
<td>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.</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>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

- 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.
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 78.
- 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

CMD will show commands for this list. Type command and press ENTER
LISTS: BMCUHIST CA CK CL CO DB DS IC IM IS IX MQT MX OS PA PDD PG PL RE RI SG
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
TS QZUDA1 ASUQA QZUALL BPO  402 A
QZUDB1 ASUQA SYSEFLET BPO 1346                        E
QZUDB2 ASUQA SYSEFLET BPO 1343                        E
QZUDB3 ASUQA SYSEFLET BPO 1347                        E
QZUDB4 ASUQA SYSEFLET BPO 1348                        E
QZUDB5 ASUQA SYSEFLET BPO 1350                        E
QZUDB6 ASUQA SYSEFLET BPO 1351                        E
QZUDCF ASUQA SYSEFLET BPO  615 A
QZUDCI ASUQA SYSEFLET BPO 1352                        E
QZUDCI15 RDAHZE3 SYSEFLET BPO 1377                        E
QZUDCI19 ASUQA SYSEFLET BPO 1378                        E
QZUDCI01 ASUQA SYSEFLET BPO 1355                        E
QZUDCI02 ASUQA SYSEFLET BPO 1356                        E
QZUDCI03 ASUQA SYSEFLET BPO 1358                        E
QZUDCI04 ASUQA SYSEFLET BPO 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

CMD will show commands for this list. Type command and press ENTER
LISTS: AL BMCUHIST CA CK CL CO DB DS FK IC IM IS IX LK MQT MX NP OS PA PDD PG PL
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 BPO       0    1      130        A    A
QZUDA1.QZUS02A1   ASUQA     16 BPO       0    2       146        A    A
QZUDA1.QZUS03A1   ASUQA      0 BPO       4    1     1440        A    A
QZUDA1.QZUS04A1   ASUQA     64 BPO       0    2      540        A    A
QZUDA1.QZUS05A1   ASUQA      0 BPO       4    1      720        A    A
```
Generating secondary lists

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

   ![Figure 7](image)

2. Press Enter.

   CATALOG MANAGER first displays a secondary list of tables within the first source table space (Figure 8).
Figure 8  Table List panel for first source table space

```
DEFF-R --------------------------  TABLE LIST  ------------------ ROW 1 OF 1
Command ===>                                                  Scroll ====> CSR
03
CMD will show commands for this list. Type command and press ENTER
Lists: AL CA CD CK CL CO CP C2 DB DS DT FK IC IS IX KC KU LK MK MQT MX NP
QUALIFIER: TABLESPACE=QZUDA1.QZUS01A1
C   Table Name                     Database Tblspace ColsPK Type  Rows  Pages
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
QZU.QZUT01_DA1S01              QZUDA1   QZUS01A1  21  1 T     2036    128
******************************  BOTTOM OF DATA  ******************************
```

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

```
DEFF-R --------------------------  TABLE LIST  ------------------ ROW 1 OF 2
Command ===>                                                  Scroll ====> CSR
03
CMD will show commands for this list. Type command and press ENTER
Lists: AL CA CD CK CL CO CP C2 DB DS DT FK IC IS IX KC KU LK MK MQT MX NP
QUALIFIER: TABLESPACE=QZUDA1.QZUS02A1
C   Table Name                     Database Tblspace ColsPK Type  Rows  Pages
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
QZU.QZUT01_DA1S02              QZUDA1   QZUS02A1  21  1 T     2036    128
QZU.QZUT02_DA1S02              QZUDA1   QZUS02A1   4  0 T        0      0
******************************  BOTTOM OF DATA  ******************************
```

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>IX QZU.QZUT01_DA1S02</th>
<th>QZUDA1</th>
<th>QZUS02A1</th>
<th>21</th>
<th>T</th>
<th>2036</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>KC QZU.QZUT02_DA1S02</td>
<td>QZUDA1</td>
<td>QZUS02A1</td>
<td>4</td>
<td>0</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).
2 Press END.

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

Figure 12  Column List panel for second source table

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 H, “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.
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.

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.
### Issuing Wait-for-Enter commands against multiple objects

#### Chapter 2 Getting started with CATALOG MANAGER

**Figure 15** Tablespace List panel with shortcut commands for copying multiple table spaces

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DEFF-R</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TABLESPACE LIST</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ROW 1 OF 5</strong></td>
<td></td>
<td>-----</td>
</tr>
</tbody>
</table>

CMD will show commands for this list. Type command and press ENTER

Lists: AL BMCUHIST CA CL CO DB DS FK IC IM IS IX LK MMT MX NP OS PA PDD PG PL
QUALIFIER: DATABASE=QZUDA1

<table>
<thead>
<tr>
<th>Cmd</th>
<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>02</td>
<td>BMCCOPY</td>
<td>ASUOA</td>
<td>4</td>
<td>BP0</td>
<td>0</td>
<td>1</td>
<td>130</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>02</td>
<td>QZUDA1.QZUS02A1</td>
<td>ASUOA</td>
<td>16</td>
<td>BP0</td>
<td>0</td>
<td>2</td>
<td>146</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>02</td>
<td>QZUDA1.QZUS03A1</td>
<td>ASUOA</td>
<td>0</td>
<td>BP0</td>
<td>4</td>
<td>1</td>
<td>1440</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>02</td>
<td>QZUDA1.QZUS04A1</td>
<td>ASUOA</td>
<td>64</td>
<td>BP0</td>
<td>0</td>
<td>2</td>
<td>540</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>02</td>
<td>QZUDA1.QZUS05A1</td>
<td>ASUOA</td>
<td>0</td>
<td>BP0</td>
<td>4</td>
<td>1</td>
<td>720</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

**** Bottom of DATA ****

#### 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

<table>
<thead>
<tr>
<th>CMD</th>
<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></td>
<td>QZUDA1.QZUS01A1</td>
<td>ASUQA</td>
<td>4 BPO</td>
<td></td>
<td>1</td>
<td>130</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>QZUDA1.QZUS02A1</td>
<td>ASUQA</td>
<td>16 BPO</td>
<td></td>
<td>2</td>
<td>146</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>QZUDA1.QZUS03A1</td>
<td>ASUQA</td>
<td>0 BPO</td>
<td></td>
<td>4</td>
<td>1440</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>QZUDA1.QZUS04A1</td>
<td>ASUQA</td>
<td>64 BPO</td>
<td></td>
<td>2</td>
<td>540</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>QZUDA1.QZUS05A1</td>
<td>ASUQA</td>
<td>0 BPO</td>
<td></td>
<td>4</td>
<td>720</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### Figure 17  Tablespace List panel with exclude commands

<table>
<thead>
<tr>
<th>CMD</th>
<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></td>
<td>QZUDA1.QZUS01A1</td>
<td>ASUQA</td>
<td>4 BPO</td>
<td></td>
<td>1</td>
<td>130</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X QZUDA1.QZUS02A1</td>
<td>ASUQA</td>
<td>16 BPO</td>
<td></td>
<td>2</td>
<td>146</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>X QZUDA1.QZUS03A1</td>
<td>ASUQA</td>
<td>0 BPO</td>
<td></td>
<td>4</td>
<td>1440</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>QZUDA1.QZUS04A1</td>
<td>ASUQA</td>
<td>64 BPO</td>
<td></td>
<td>2</td>
<td>540</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>QZUDA1.QZUS05A1</td>
<td>ASUQA</td>
<td>0 BPO</td>
<td></td>
<td>4</td>
<td>720</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---
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.

**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>

For more information about enabling Fast Path Navigation, see the Installation System User Guide and the BMC Products and Solutions for DB2 Configuration Guide.
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
Manipulating plans ....................................................... 70
Setting the MEMLIMIT system parameter ............................ 70
Using options to control your environment ........................... 70
  Using the installation options ....................................... 71
  Using the user options ............................................... 72
  Using the product options ........................................... 72
  Putting it all together ................................................ 74
Setting user options ...................................................... 75
  Defining an options data set ........................................ 76
  Setting basic options ............................................... 78
  Setting general options ............................................. 81
  Setting object use options .......................................... 84
  Setting JCL Generation options ..................................... 85
  Setting data set options ............................................. 87
  Setting SQL and confirm options ................................... 89
  Setting SQL SELECT options ....................................... 92
  Setting panel graphic options ...................................... 93
  Setting CATALOG MANAGER switches .............................. 94
  Setting DESCRIBE options ......................................... 97
Setting product options ................................................ 98
  Setting the JCL options for job cards .............................. 100
  Setting the JCL options for STEPLIBs .............................. 103
  Setting the JCL options for static data sets ....................... 104
  Setting the JCL options for tapes .................................. 109
  Setting the JCL options for temporary work data sets .......... 111
  Setting the JCL options for permanent data sets ................. 114
  Setting the JCL generation data group options ................... 119
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
- Schema
- Current Debug Mode
- Current DECIMAL Rounding Mode
- Current Routine Version

**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>
<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>
<tr>
<td>ACTvrDE</td>
<td>Data Editing and Browsing</td>
<td>enables access to data editing and browsing functions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This plan does not override DB2 table authorizations.</td>
</tr>
<tr>
<td>ACTvrDG</td>
<td>Generate SQL</td>
<td>allows SQL statement generation</td>
</tr>
<tr>
<td>ACTvrDH</td>
<td>Utility Status Display</td>
<td>enables displaying the status of BMC utilities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>ACTvrDK</td>
<td>Command Generation and Execution</td>
<td>enables generating and executing DB2 operator commands</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Grant EXECUTE authority on this plan to users who should be able to issue DB2 operator commands, such as START, STOP, DISPLAY, and TERM.</td>
</tr>
<tr>
<td>ACTvrDL</td>
<td>Log Table Maintenance</td>
<td>allows browsing rows in the CATALOG MANAGER Audit, Session, and Drop Recovery Logs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>ACTvrDM</td>
<td>Display DB2 Catalog and SQL Information</td>
<td>displays information about the DB2 catalog and SQL generation and execution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This plan allows the minimum access that is required to use CATALOG MANAGER. The ACTvrDM plan also allows you to execute a worklist through the Execution component.</td>
</tr>
<tr>
<td>ACTvrDS</td>
<td>Search</td>
<td>enables search functions</td>
</tr>
<tr>
<td>ACTvrDU</td>
<td>Grant Authorities and Submit BMC Utilities or IBM Utilities</td>
<td>enables generating and submitting JCL for BMC and IBM utilities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Grant EXECUTE authority on this plan to users who should be able to grant authorities or submit BMC or IBM utilities.</td>
</tr>
</tbody>
</table>
Manipulating plans

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.

### 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 F, “JCL Generation product options.”

### 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.
Using the installation options

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 System User Guide* and the *BMC Products and Solutions for DB2 Configuration 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 Appendix E, “CATALOG MANAGER installation options.”
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 R after the option value and then enclose the value in parentheses, as shown in the following example:

```
DBCS=(N,R),
```

**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 75. For more information about refreshing user options, see the *Installation System User Guide* and the *BMC Products and Solutions for DB2 Configuration Guide*.

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 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 98. For more information about POFs, see Appendix F, “JCL Generation product options.”
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 13  Options panels (Part 1 of 2)

<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

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.

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>

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.

If you want to use the default values, type NONE. If you do not specify an options data set, your options will not be saved.

**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 In the **Action** field, type 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).

Figure 21 Allocate Data Set

```
DEFF-R ------------------------ Allocate Data Set -------------------------
Command ===>  
Allocate data set with the following values N (Y/N)  

Data Set Name: RDACRJ1.BMCCAT.USEROPT 
DD Name . . . :  

    Volume Serial . . . . . . (Blank for authorized default volume) * 
    Generic Unit . . . . . . SYSLDDA (Generic group name or unit address) * 
    Space Units . . . . . . CYLS (BLKS, TRKS or CYLS) 
    Primary Quantity . . . 2 (In above units) 
    Secondary Quantity . . 1 (In above units) 
    Directory Blocks . . . 30 (Zero for sequential data set) 
    Record Format . . . . . . VB 
    Record Length . . . . . 4092 
    Block Size . . . . . . . 4096 
    Expiration Date . . . . (MM/DD/YYYY)  
```

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  1 to 18 of 18
Command ===>                                                  Scroll ===> PAGE

User options Dsn . . . . . .   RDACRJ.BMCCAT.USEROPT(ACTUSROP)
Describe pds member . . . . .   RDACRJ
DB2 VCAT . . . . . . . . . . .   DEFFCAT
Default SQLID . . . . . . . .   
Max lines per list . . . . .   0  0-9999, 0-Unlimited
Maximum # of select lines . .  300  0-9999, 0-Unlimited
Profile . . . . . . . . . . .   

1) Edit General options . . .  N  Y/N Edit General options
2) Edit Object selections . . .  N  Y/N Edit Mixed list and HDDL options
3) Edit JCL Generation options  N  Y/N Edit JCLGEN and POF options
4) Edit Dataset names . . . . .  N  Y/N Edit Dataset names
5) Edit SQL and Confirm options N  Y/N Edit SQL and Confirm panel options
6) Edit SQL Select options . . .  N  Y/N Edit SQL Select options
7) Edit Color options . . . . .  N  Y/N Edit Color settings
8) Edit Switches . . . . . . .   N  Y/N Edit Switches
9) Edit Describe options . . .  N  Y/N Edit User Describe options
```

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

Figure 23  Options Dataset has changed panel

|------------------------- Options Dataset has changed -----------------------|
| Command ===> |
| Your CATALOG MANAGER Options data set has changed. |
| Options dataset:  RDACRJ.BMCCAT.USEROPT(ACTUSR) |
| Type NONE for the data set name if you want to use default values. Any option that you set will not be saved. To save your options, specify an options data set. |
| Action . . . . : N (Y/N) |
| Type Y if you want to load your options from the specified data set. If you have defined the data set on another ID and you want to use that data set, specify the name of the data set. |
| Type N if you have not created a PDS data set or member for your options. The new data set or member uses the current settings. If the PDS does not exist, you will be prompted to create it. |

- 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 97.

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.
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 12, “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.
3 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>AUBDYN1</th>
<th>General Options</th>
<th>1 to 10 of 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>General Options</td>
<td>Scroll PAGE</td>
</tr>
<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>0-199, 0 - unlimited</td>
<td></td>
</tr>
<tr>
<td>DB2 feedback size . . .</td>
<td>16-999, Size of feedback area in KB</td>
<td></td>
</tr>
<tr>
<td>Truncation chars . . .</td>
<td>Long name truncation character</td>
<td></td>
</tr>
<tr>
<td>Truncation position . . L</td>
<td>L-Left, M-Middle, R-Right</td>
<td></td>
</tr>
<tr>
<td>Char field max width . .</td>
<td>10-99 For list display</td>
<td></td>
</tr>
<tr>
<td>Terse level . . . . VERBOSE</td>
<td>TERSE/VERBOSE</td>
<td></td>
</tr>
</tbody>
</table>
```

4 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.

5 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 198.

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

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.
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>N</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 98.

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.
Setting data set options

- 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**

```
Datasets                          1 to 11 of 11

Command ===>                  Scroll ===> PAGE

Print Dsn. . . . . . . . . . . . &ZUSER..BMCCAT.PRINT
Catalog Manager Work Dsn . . . &ZUSER..BMCCAT.WORK
SQL Output Dsn . . . . . . . .  &ZUSER..BMCCAT.SQL
Online Bind default DBRM Dsn . 'BMCACT.V71S2.DBRM'
User Utilities Profile Dsn . .
JCL Generation Dsn . . . . . . &ZUSER..BMCCAT.JCL()
```
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.

To create a user utility profile data set, see page 293.

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.
Setting SQL and confirm options

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</th>
<th>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>Auto-commit... Y Y/N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate SQL/DSN... N Y/N Default value for Generate/Create</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Confirm SQL and DSN panel defaults</td>
<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>
<td></td>
</tr>
<tr>
<td>Execute... N Y/N Drop/Revoke always default to N</td>
<td></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>
<td></td>
</tr>
<tr>
<td>Drop recovery and revoke reassign defaults</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add dependency list... N Y/N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drop recovery on... N Y/N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log image copies... N Y/N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add cascade report... N Y/N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plan table... PLAN_TABLE</td>
<td></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.

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).

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

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

9 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**).
Setting CATALOG MANAGER switches

Figure 30  Colors panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Colors</th>
<th>1 to 11 of 11</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>RED</td>
<td>Enter ? for choices</td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td>GREEN</td>
<td>Enter ? for choices</td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td>BLUE</td>
<td>Enter ? for choices</td>
<td></td>
</tr>
<tr>
<td>Pink</td>
<td>PINK</td>
<td>Enter ? for choices</td>
<td></td>
</tr>
<tr>
<td>Yellow</td>
<td>YELLOW</td>
<td>Enter ? for choices</td>
<td></td>
</tr>
<tr>
<td>Turq</td>
<td>TURQ</td>
<td>Enter ? for choices</td>
<td></td>
</tr>
<tr>
<td>White</td>
<td>WHITE</td>
<td>Enter ? for choices</td>
<td></td>
</tr>
<tr>
<td>Highlight</td>
<td>blank, REVERSE, USCORE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Set DASD MANAGER Graphics Y/N Edit DASD Graphics options

4 For the various colors, type the default color.

5 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.

6 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.

7 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.

To access the Switches panel, follow the steps in “To set switches” on page 95.

**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 ====&gt;</th>
<th>Switches</th>
<th>1 to 6 of 6</th>
<th>Scroll ====&gt;</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Server...</td>
<td>N</td>
<td>DBCS ...</td>
<td>N</td>
<td>HDDL Auths...</td>
</tr>
<tr>
<td>DEBUG...</td>
<td>N</td>
<td>Label ...</td>
<td>N</td>
<td>Shared data...</td>
</tr>
<tr>
<td>Cmp &gt; 32k...</td>
<td>Y</td>
<td>Drop ...</td>
<td>Y</td>
<td>SQL flow ...</td>
</tr>
<tr>
<td>Last used prof</td>
<td>N</td>
<td>Revoke BY...</td>
<td>N</td>
<td>Server SSID...</td>
</tr>
<tr>
<td>SQL Ownerid...</td>
<td>.</td>
<td>SQL</td>
<td>SQL/TSO</td>
<td></td>
</tr>
<tr>
<td>Build SQLID before GRANT...</td>
<td>N</td>
<td>Y/N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dev debug...
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</td>
</tr>
<tr>
<td></td>
<td>be generated only if the current SQLID has SYSADM or SYSCTRL authority. If</td>
</tr>
<tr>
<td></td>
<td>the current SQLID does not have SYSADM or SYSCTRL authority and is not the</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>the upper left corner</td>
</tr>
<tr>
<td>_ Wild</td>
<td>for table lists, consider an underscore as a wildcard character if no other</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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></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>C</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 204.

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.
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 - MAIN MENU V11.01.00 ------------------
COMMAND ===> 
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
(C) Copyright 2013 BMC Software, Inc.
All Rights Reserved
```

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 100</td>
</tr>
<tr>
<td>Steplib Options</td>
<td>“Setting the JCL options for STEPLIBs” on page 103</td>
</tr>
<tr>
<td>Static Data Set Options</td>
<td>“Setting the JCL options for static data sets” on page 104</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 111</td>
</tr>
<tr>
<td></td>
<td>■ “Setting the JCL options for permanent data sets” on page 114</td>
</tr>
</tbody>
</table>
Setting the JCL options for job cards

### Option

<table>
<thead>
<tr>
<th>Option</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation Data Group Options (GDGs)</td>
<td>“Setting the JCL generation data group options” on page 119</td>
</tr>
<tr>
<td>Debugging, Display and Execution Options</td>
<td>“Setting the JCL debugging, display, and Execution options” on page 121</td>
</tr>
<tr>
<td>Utility Options</td>
<td>“Setting the JCL utility installation options module name options” on page 124</td>
</tr>
<tr>
<td>Listdef and Template Data Sets</td>
<td>“Setting the LISTDEF and TEMPLATE data set options” on page 130</td>
</tr>
<tr>
<td>User Defined Variable Values</td>
<td>“Setting user variables” on page 131</td>
</tr>
</tbody>
</table>
| Product Options File (POF) Functions        | ■ “Creating a user POF” on page 133   
■ “Updating a user POF” on page 134  
■ “Using multiple POFs” on page 135  
■ “Refreshing the initial POF” on page 136  
■ “Generating POF reports” on page 137  
■ “Reusing a POF in a subsequent installation” on page 138  
■ “Overriding POF values in SLIBs” on page 139  
■ “Adding steps to the JCL” on page 141  
■ “Obtaining a list of TEMPLATEs or LISTDEFs in CATALOG MANAGER” on page 144 |

6 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.
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........................................
Region size ................................ OM (See JCL Reference for valid options)
Memlimt ........................................ 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 ===>
Type data and press Enter.
DSNEXIT . . . . . . . . SYS3.DBDC.DSNEXIT
DB2 DSNLOAD . . . . . . SYS2.DB2V10M.DSNLOAD
Override lib. . . . . .
CATALOG MANAGER . . . .
ALTER/CHANGE MANAGER . .
DASD MANAGER PLUS . . .
EXECUTION . . . . . . . 'BMACADMN.V101.10.BMCLINK'
COPY PLUS . . . . . . .
REORG PLUS . . . . . .
LOADPLUS . . . . . . .
UNLOAD PLUS . . . . . .
RECOVER PLUS . . . . .
CHECK PLUS . . . . . .
SQL EXPLORER . . . . .
Additional lib . . . .
IOA LOAD 1 . . . . . .
IOA LOAD 2 . . . . . .
```
Setting the JCL options for static data sets

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.)
Include SYSPRIN2 DD........ N   (Y/N)

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 381.
Setting the JCL options for static data sets

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>
<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 118.</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.
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.

6 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>ddname</th>
<th>Used in JCL cleanup by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discard (SYSDISC)(^a)</td>
<td>SYSD(mm)</td>
<td>BMC LOADPLUS \nbm LOAD</td>
</tr>
<tr>
<td>Error</td>
<td>SYSER(mm)</td>
<td>BMC CHECK PLUS \nbm LOADPLUS IBM LOAD DATA IBM LOAD</td>
</tr>
<tr>
<td>Map</td>
<td>SYSMAP</td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Punch(^b)</td>
<td>SYSPUNCH</td>
<td>BMC REORG PLUS \nbm REORG</td>
</tr>
<tr>
<td>Unload (SYSREC)(^b)</td>
<td>SYSR(mm) or R(nnnnyyyy)</td>
<td>BMC REORG PLUS \nbm REORG \n\tThe 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>

\(^a\) 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.

\(^b\) 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.
Setting the JCL options for static data sets

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>SORTOnnn</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>IBM 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.

The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

9 In the **Include SYSPRIN2 DD** field, type **Y** or **N** to specify whether to generate the `//SYSPRIN2 DD SYSOUT=*` DD in the JCL.

The SYSPRIN2 output data set contains SYSPRINT output messages for versions 10.2 and later of the following BMC utilities:

- CHECK PLUS
- LOADPLUS
- REORG PLUS
- UNLOAD PLUS

If you type **Y**, you can view the SYSPRINT output from a utility while an execution job runs the utility or when an execution job cancels during the running of the utility.
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>

### NOTE

SYSPRIN2 data sets have the following restrictions:

- When you specify BMCSTATS YES or UPDATEDB2STATS YES for LOADPLUS or REORG PLUS, SYSPRIN2 does not contain the statistics report from the Common Statistics component.
- When invoking the IBM DSNUTILB utility, REORG PLUS and LOADPLUS ignore the SYSPRIN2 DD statement.

10. Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel.
Figure 37  JCL Generation Tape Options Update panel

---

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.

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.
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.

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**

   ----------------- JCL GENERATION OPTIONS FOR SORT FILES UPDATE -----------------
   COMMAND ===> 
   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)
   $ORTPARM 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.

The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

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.

The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

5 If you typed N in the **Data set sizing option** field in step 2 on page 105, 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 **SORTPARAM 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, FCOPYDD, or EXPORTDDN)
- 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.

**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.

Consider the following items when you specify the prefix:

- JCL Generation automatically appends the ddname to the prefix to create the name of the data set.

To suppress the ddname, specify Y for the appropriate keyword in the POF (Table 20).

Table 20 POF keywords used to suppress the ddname

<table>
<thead>
<tr>
<th>Data set</th>
<th>POF keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>flashcopy</td>
<td>FCPY_SUPPRESS_SUFF</td>
</tr>
<tr>
<td>local primary copy</td>
<td>PCPY1_SUPPRESS_SUFF</td>
</tr>
<tr>
<td>local backup copy</td>
<td>PCPY2_SUPPRESS_SUFF</td>
</tr>
<tr>
<td>recovery primary copy</td>
<td>RCPY1_SUPPRESS_SUFF</td>
</tr>
<tr>
<td>recovery backup copy</td>
<td>RCPY2_SUPPRESS_SUFF</td>
</tr>
<tr>
<td>primary SYSREC</td>
<td>UNLD1_SUPPRESS_SUFF</td>
</tr>
<tr>
<td>backup SYSREC</td>
<td>UNLD2_SUPPRESS_SUFF</td>
</tr>
</tbody>
</table>

- 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).

Figure 40 Using the &GDG symbolic variable

--- JCL GENERATION DATA SET OPTIONS FOR LOCAL PRIMARY COPY UPDATE ---

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

Enter Data Set Prefix below:

. . &PREFIX.&OBNOD(&GDG)

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 . . . . . . . . . (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)

When 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

```plaintext
//**  UTILITY COPY DD STATEMENTS
/**---------------------------------------------------------------
//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),
  // 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)
/**---------------------------------------------------------------
```
3 In the **Unit Name** field, type the name of the unit.

Consider the following items when you specify the unit:

- The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

- 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 104), 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.

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</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 -</td>
<td>Generation Data Group Options (GDGs)</td>
</tr>
<tr>
<td>Main Menu</td>
<td></td>
</tr>
</tbody>
</table>

Figure 42  JCL Generation Generation Data Group Options Update panel

```
----------- JCL GENERATION GENERATION DATA GROUP OPTIONS UPDATE -----------
COMMAND ===>
Type data and press Enter.
Define GDG base at JCL generation?  .  N  (Y/N)
Specify NSCR on GDG definition? .  N  (Y/N)
Number of primary copy GDG entries . 10  (1-255)
Number of recovery copy GDG entries . 10  (1-255)
Type GDG Model data set below:
.  .  SYS1.MODEL
NOTE: GDGs only apply to copy data sets.
```

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>
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.

A 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 dsso/cc, where dsso is the data set sizing option and cc is a comment code. Table 21 lists the comment codes that Execution generates in the JCL.
Table 21  Comment codes for data set sizing

<table>
<thead>
<tr>
<th>Data set sizing options</th>
<th>Comment code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B, C, or O</td>
<td>C</td>
<td>uses statistics from the DB2 catalog</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>uses the high relative-byte address (RBA)</td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>uses multiple objects to size one data set (for example, SYSUTs)</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>indicates that the data set could not be sized because statistics could not be found</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>uses VSAM object sampling</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>uses the following formula to calculate the SORTWK size: (work space * 2) / number of SORTWK data sets</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>warns that the sizing might be inaccurate</td>
</tr>
</tbody>
</table>

Figure 44 shows example comments in the Execution JCL.

B  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.
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.

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 Specify a Plan name to run DSNTIAD field, type the name of the DB2 plan to run the IBM DSNTIAD program.

6 In the Pre Job Step JCL INCLUDE member name field, type the name of a JCL member to be included before each step in the JCL.

7 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.

8 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.

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

10 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 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>Utility Options</td>
</tr>
<tr>
<td>Utility Options</td>
<td>BMC Utility Option Module Names</td>
</tr>
</tbody>
</table>

Figure 45  JCL Generation BMC Utility Option Module Names Update panel

```
COMMAND ===>
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.

### NOTE
CATALOG MANAGER currently does not use the values on the Online Reorg Utility Options panel.

#### 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**

```
COMAND =====>

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. ALTER and CHANGE MANAGER use this value when reorganizing a table space by using an online reorganization (SHRLEVEL CHANGE).

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.

ALTER and CHANGE MANAGER use this value when reorganizing a table space by using an online reorganization (SHRLEVEL CHANGE).

The name can be from 1 to 72 characters long, and can contain the &ZUSER or &USERID symbolic variable.

---

**NOTE**

You can indicate whether to include the name of the mapping table in the syntax for the IBM REORG utility on the ALTER and CHANGE MANAGER Analysis Options panel.

The REORG PLUS utility invokes the IBM DSNUTILB utility control program to enable certain features. If you have specified to use the REORG PLUS utility, you still need to specify mapping table information. For information about the features for which REORG PLUS invokes DSNUTILB, see the **REORG PLUS for DB2 Reference Manual**.

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. You can use the PROC for the product instead of direct program invocations for standard JCL.
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</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Primary Menu</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>Utility Options</td>
</tr>
<tr>
<td>Utility Options</td>
<td>Non worklist JCL PROC Options</td>
</tr>
</tbody>
</table>

---

**Figure 47  PROC and STEP Names panel**

```
COMMAND ===> PROC AND STEP NAMES

Use JCL Procedures (PROCS) for standard JCL?   Y (Y or N)
Generate SET variables in JCL? ............ Y (Y or N)

Function  Procname  Procstep  Function  Procname  Procstep
BMC REORG  BMCREORG  RSTEP      TSO BATCH
BMC COPY   DSNUTILB
BMC LOAD   DSNICOPY
BMC UNLOAD  IDCAMS
BMC RECOVER  IEFBR14
BMC CHECK  BMC STATS
BMC TRIG   BMC CPRS
BMC UPRS   BMC STOP
USER DEF
```

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. You are responsible for creating the PROC.
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 specify the PDS that contains the PROCs in the Jcllib field on the JCL Generation Jobcard Options panel (see “Setting the JCL options for job cards” on page 100). To add JCLLIBs to the concatenation, modify the AJXJCLU SLIB. Alternatively, you can include the JCLLIB or PROCLIB in your jobcard JCL.

You can also specify to use PROCs for standard JCL in the PROC_USE POF keyword in your POF.

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

Alternatively, you can define the name of a PROC in the following POF keywords in your POF:

- PROC_BMCCHECK_NAME =
- PROC_BMCCOPY_NAME =
- PROC_BMCCPRS_NAME =
- PROC_BMCCLOAD_NAME =
- PROC_BMCRECOVER_NAME =
- PROC_BMCREORG_NAME =
- PROC_BMCSTATS_NAME =
- PROC_BMCUNLOAD_NAME =
- PROC_BMCUPRS_NAME =
- PROC_DSNUTILB_NAME =
- PROC_DSN1COPY_NAME =
- PROC_IDCAMS_NAME =
- PROC_IDFCRS_NAME =
- PROC_IEFBR14_NAME =
- PROC_TSO_NAME =
Setting the LISTDEF and TEMPLATE data set options

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

Alternatively, you can define the name of a STEP in the following POF keywords in your POF:

- PROC_BMCSTOP_NAME = PROC_USER_DEFINED =
- PROC_BMCTRIG_NAME =

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</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td>Primary Menu</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>Listdef and Template Data Sets</td>
</tr>
</tbody>
</table>
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 SYSEMPL.

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 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>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>

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.

---

**NOTE**
You can update all of the ISPF variables in the ISPF profile with the variables in the POF by using one of the following **Command** line commands:

- To use the variables in the initial POF, enter `TSO POFRESET`.
- To use the variables in a specified initial or user POF, enter `TSO POFRESET POF(dataSetName(POFMember))`.

To enable these commands, you must modify the POFRESET CLIST. The CLIST is located in the \*HLQ,DBCLIB library. A current copy of this CLIST must be in the same SYSPROC concatenated library as your other CLISTs.

---

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 133 to create an additional POF.

2. On the Command line, enter TSO POFRESET POF(dataSetName(POFMember)).

To specify the use of a different user POF, 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.

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.
Figura 53  Specifying the prefix for a copy 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.

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

![JCL Generation Jobcard Options Update panel](image)

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**

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

3 From the JCL Generation Update - Main Menu, select **Debugging, Display and Execution Options** and press Enter.
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

---

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:
SYNCDLT . . . N (Y/N) BINDFAIL . . . N (Y/N)
HASFFAIL . . . N (Y/N) HSIWARNRC . . . (NUMERIC)
REBINDFAIL . . . N (Y/N) REBINDRC . . . (NUMERIC)
2MEGSQL . . . N (Y/N) NOFAILNOIMAGECOPY 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

```*
//*--------------------------------
//* END OF JOBSTEP
//*--------------------------------
//*--------------------------------
//* END OF STEP INCLUDE MEMBER
//*--------------------------------
// INCLUDE MEMBER=STEPEND
```
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 306.

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.
You cannot modify ACTCOMND. However, you can modify the ACTCOMNU user commands table that BMC provides in HLQ.CNTL. The user commands table might contain commands for invoking the IBM DB2 data editor, modifications to existing commands, and any new commands. Your user commands table overrides ACTCOMND. When you start CATALOG MANAGER, the product merges the primary commands table with the user commands table that you specify in the UCOMMD installation option.
Setting up and modifying the user commands table

Before you use your user commands table, you must set it up. You can also modify the table.

**Setting up the user commands table**

Perform the following steps to set up the table:

1. Copy `HLQ.CNTL(ACTCOMNU)` to `HLQ.UDBCNTL`.
2. *(optional)* Rename `ACTCOMNU`.
3. Specify the name of your user commands table in the UCOMD installation option.
4. *(optional)* Copy the commands that you want to modify from the `ACTCOMMD` member and paste them into `ACTCOMNU`.
5. *(optional)* Modify your user commands table by editing, disabling, or adding commands.
6. Run the `USRCOMND` job in the `HLQ.JCL` or `HLQ.UDBCNTL` data set to compile and link your user commands table.

**Editing commands in the user 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. For information about the syntax and parameters used in the commands table, see “Commands table syntax and parameters” on page 149.

**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.
### Setting up and modifying the user commands table

#### Figure 62 STATUS command

```plaintext
*STATUS
$ACTCMD STATUS,#STU,LOAD=ACTXXXXX,HELP=ACTHSTAU,CAT=YES, UTILCMD=YES
```

#### Figure 63 DISPLAY UTILITY command

```plaintext
*DISPLAY UTILITY
$ACTCMD DISUTILITY,#DSU,LOAD=ACTXXXXX,HELP=ACTHKDUT,CAT=YES, UTILCMD=YES,OBJECTS=(DB+IC+IM+IX+SG+TB+TS)
```

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

```plaintext
*DISPLAY STATUS
$ACTCMD STATUS,#DSU,LOAD=ACTXXXXX,HELP=ACTHKDUT,CAT=YES, UTILCMD=YES,OBJECTS=(DB+IC+IM+IX+SG+TB+TS)
```

3. Save the changes.

4. Run the USRCOMND job in the HLQ.JCL or HLQ.UDBCNTL data set 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.

   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.
Setting up and modifying the user commands table

4 Run the USRCOMND job in the HLQ.JCL or HLQ.UDBCNTL data set to compile and link your user commands table.

Disabling commands in the user commands table

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 in the HLQ.JCL or HLQ.UDBCNTL data set 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 149.

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 149.

3 Write the CLIST. For more information, see “Writing user commands as CLISTs” on page 153.

4 Run the USRCOMND job in the HLQ.JCL or HLQ.UDBCNTL data set to compile and link the user commands table.
Commands table syntax and parameters

Entries in the commands table use the following syntax:

Table 22 Syntax of commands table entry

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

Table 23 describes the variables in the commands table syntax.

Table 23 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>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 of</td>
</tr>
<tr>
<td></td>
<td>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. Also, some of the parameters in the member are described in this</td>
</tr>
<tr>
<td></td>
<td>section.</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 39, 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 `excoobjc` 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).
&WFEK

This parameter indicates whether this command is a Wait-for-Enter command. A Wait-for-Enter command is one that you can specify for multiple items in a list so that CATALOG MANAGER can process them as a group when you press the Enter key. If &WFEK=YES, all objects found with the repeat function (equal sign) or the ALL parameter are added to the object list and one call is made to the user program. If &WFEK=NO (the default), the REPEAT function is not allowed, and CATALOG MANAGER displays individual user prompts for each specified object. For CLISTs, use WFEK=NO.

Migrating a user commands table to a new release of CATALOG MANAGER

During product installation, you can copy your source user commands table from an existing library to a new library. This action 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 Compare HLQ.DBCNTL(ACTCOMNU) for the previous release to HLQ.DBCNTL(ACTCOMNU) for the latest release.

2 Perform one of the following actions:
   - If differences exist between the two members, copy HLQ.CNTL(ACTCOMNU) for the latest release to the latest release of the HLQ.UDBCNTL library. Copy the modifications that you made to HLQ.UDBCNTL(ACTCOMNU) to member ACTCOMNU in the latest release of the HLQ.UDBCNTL library.
   - If no differences exist between the two members, copy HLQ.UDBCNTL(ACTCOMNU) for the previous release to the latest release of the HLQ.UDBCNTL library.

**NOTE**

This step assumes that you have made your changes in the HLQ.UDBCNTL library.

3 (optional) Make any additional changes to the user commands table source.

4 Run the USRCOMND job in the HLQ.JCL or HLQ.UDBCNTL data set 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 24 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>ACTCOMNU</td>
<td>user commands table</td>
</tr>
<tr>
<td>ACTUSER</td>
<td>sample user command program in ASM</td>
</tr>
<tr>
<td>$ACTEXEC</td>
<td>macro DSECT for user program variables</td>
</tr>
<tr>
<td>$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>
</tbody>
</table>

To view $ACTVARS, type **CLIST ACTVARS** in the **Cmd** column of a list, and then press **Enter**. The command displays a panel that shows the value of all variables.

USRCOMND    | JCL to assemble and link the ACTCOMND source code for the commands 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 25 lists the key parameters.

Table 25  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></td>
<td>The options are as follows:</td>
</tr>
<tr>
<td></td>
<td>- E - END</td>
</tr>
<tr>
<td></td>
<td>- R - return</td>
</tr>
<tr>
<td></td>
<td>- C - cancel</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</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></td>
<td>If PARSE does not equal YES, the value of this parameter is the same as the</td>
</tr>
<tr>
<td></td>
<td>value of the EXCCOBJL parameter.</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 26.
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 26 Parameters for use with the $ACTULOG macro

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pointer</td>
<td>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.</td>
</tr>
<tr>
<td>OPT</td>
<td>specifies which logs to write to. The default value is S. The options are as follows:</td>
</tr>
<tr>
<td></td>
<td>A - Audit Log</td>
</tr>
<tr>
<td></td>
<td>S - Session Log</td>
</tr>
<tr>
<td></td>
<td>B - both the Audit Log and the Session Log</td>
</tr>
<tr>
<td>FUNC</td>
<td>address of a field containing the 16-byte function to log</td>
</tr>
<tr>
<td></td>
<td>It does not need to match the command name. If this parameter is not used, blanks are logged.</td>
</tr>
<tr>
<td>RC</td>
<td>identifies the register containing the return code to be logged</td>
</tr>
<tr>
<td></td>
<td>If this parameter is not used, X'00' is logged as the return code.</td>
</tr>
<tr>
<td>TYPE</td>
<td>address of a 10-byte value to be logged as the object type</td>
</tr>
<tr>
<td></td>
<td>If this parameter is not used, blanks are logged.</td>
</tr>
<tr>
<td>QUAL</td>
<td>address of an 8-byte value to be logged as the object name qualifier</td>
</tr>
<tr>
<td></td>
<td>If this parameter is not used, blanks are logged.</td>
</tr>
<tr>
<td>NAME</td>
<td>address of an 18-byte value to be logged as the object name</td>
</tr>
<tr>
<td></td>
<td>If this parameter is not used, blanks are logged.</td>
</tr>
</tbody>
</table>
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 27 lists the object types and their corresponding name fields.

**Table 27  Passing object types and names in user-written commands**

<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>EXCOFNDB (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 name</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>EXCOFNDB (DB name)</td>
</tr>
<tr>
<td>DBRM</td>
<td>DM</td>
<td>NA</td>
<td>DM plan name</td>
<td>DM plan</td>
<td>name</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>EXCOFNDB (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>EXCOFNID (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 name</td>
<td>IX creator</td>
<td>IX name</td>
<td>column sequence</td>
<td>NA</td>
</tr>
<tr>
<td>materialized query table</td>
<td>MQ</td>
<td>NA</td>
<td>creator</td>
<td>name</td>
<td>NA</td>
<td>EXCOFNDB (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>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>EXCOFNDB (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>EXCOFNDB (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>EXCOFNDB (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>EXCOFNDB (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 28 describes the ISPF variables that you can access from user-written commands or CLISTs.

### Table 28  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 29, set the status of the table spaces to Read Only.

The tables that are listed in Table 29 are created by CATALOG MANAGER. You can refer to the data in these tables for information on your daily operations.

### CATALOG MANAGER tables

Table 29  CATALOG MANAGER tables (Part 1 of 2)

<table>
<thead>
<tr>
<th>Table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCACm.ATTR</td>
<td>contains information about attributes</td>
</tr>
<tr>
<td>BMCACm.ATTR_VAL</td>
<td>contains the valid values for each attribute</td>
</tr>
<tr>
<td>BMCACm.AUDIT_LOG</td>
<td>contains an entry for each DDL statement that is executed by CATALOG MANAGER to modify the DB2 catalog For more information, see “Using the DDL Audit Log” on page 374.</td>
</tr>
<tr>
<td>BMCACm.CRS_VAL</td>
<td>contains cross-validation between attributes</td>
</tr>
<tr>
<td>BMCACm.DLG</td>
<td>contains dialog-specific information</td>
</tr>
<tr>
<td>BMCACm.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 29   CATALOG MANAGER tables (Part 2 of 2)

<table>
<thead>
<tr>
<th>Table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCACT\vr.EDITOR_USERS</td>
<td>contains one row for each data editing session in progress</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Clear Editor Users Table” on page 233.</td>
</tr>
<tr>
<td>BMCACT\vr.FCRRS</td>
<td>contains the Filter Combo Result Table</td>
</tr>
<tr>
<td>BMCACT\vr.GMAP</td>
<td>contains the grid mapping table</td>
</tr>
<tr>
<td>BMCACT\vr.MSG</td>
<td>contains the informational messages</td>
</tr>
<tr>
<td>BMCACT\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</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Using the Drop Recovery Log” on page 374.</td>
</tr>
<tr>
<td>BMCACT\vr.SEARCH_VARS</td>
<td>contains the values for variables that were entered in SEARCH statements and saved</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Using host variables in a search” on page 191.</td>
</tr>
<tr>
<td>BMCACT\vr.SESSION_LOG</td>
<td>contains an entry for each CATALOG MANAGER action for which logging was requested</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Using the Session Log” on page 369.</td>
</tr>
<tr>
<td>BMCACT\vr.SQL_TABLE</td>
<td>contains SQL statements that have been saved</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Working with the SQL_Table” on page 217.</td>
</tr>
<tr>
<td>BMCACT\vr.VIEW</td>
<td>contains the initial view for each utility that is supported</td>
</tr>
</tbody>
</table>
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.
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

   ![Change Access panel](image)

   The Change Access panel is displayed. (Figure 66).

   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.

<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>DEFF</td>
<td>ACT101DM</td>
<td>Call Attach</td>
</tr>
<tr>
<td>Plan</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Server</th>
<th>Current Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQLid</td>
<td>SQLid on Server</td>
</tr>
<tr>
<td>Server SSID</td>
<td>SSID of Server</td>
</tr>
<tr>
<td>Collection</td>
<td>Direct/Indirect</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

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.
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.

NOTE
As an alternative to this procedure, type SET QUALIFIER synonymQualifier on the Command line to switch catalog access.

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 “Refreshing the initial POF” on page 136.

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**
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.
Connecting to a specified SSID

Chapter 4 Accessing other DB2 subsystems

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 30 defines the parameters.

Table 30 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></td>
<td>The remote DB2 SSID must be defined in the SYSIBM.LOCATIONS table of the local DB2 SSID and must be connected by using the IBM distributed data facility (DDF).</td>
</tr>
<tr>
<td>collection</td>
<td>(optional) names the collection and package set for the remote DB2 SSID</td>
</tr>
<tr>
<td></td>
<td>The default value is the collection that the local DB2 SSID uses. If you are specifying a SQLID parameter, you can use the default value for the collection by specifying any single nonblank character (for example, ?, $, ., +, or =).</td>
</tr>
<tr>
<td>SQLID</td>
<td>(optional) specifies the current SQLID for the remote DB2 SSID</td>
</tr>
<tr>
<td></td>
<td>The default value is the SQLID that the local DB2 SSID uses.</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 RDACRJ:
CONNECT DEEG ? RDACRJ

Using the ACT101 collection for SQLID RDACRJ:
CONNECT DEEG ACT101 RDACRJ

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.
4 In the Collection field, type the name of the collection ID for the remote server.

5 Press END to display the object list.

6 On the Command line, type REFRESH.

   CATALOG MANAGER displays the objects from the remote server.

7 To generate accurate data set references when you generate JCL, refresh the POF.

   For more information, see “Refreshing the initial POF” on page 136.

8 In the Cmd column next to an object name, type DS.

9 Press Enter.

   CATALOG MANAGER displays the data set information for the object on 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 210.

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 210.

**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.
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.
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 CONTAB, see the BMC Products and Solutions for DB2 Configuration 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**

```plaintext
DEFF-S ----------------------  Connections Table  --------- Row 1 to 18 of 229
Command ===>                                                  Scroll ===> CSR
This display is for information only. No CATALOG MANAGER commands are valid.
SSID  Plan     Collection         Nickname           Location         LocID
-------------------------------------------------------------------------------
CK   D ACT101DM ACT_QA1010         CK                
DEDQ D ACT101DM ACT_QA1010         DEDQ              
DEDZ D ACT101DM ACT_QA1010         DEDZ              
DEDK D ACT101DM ACT_QA1010         DEDK              
DEDV D ACT101DM ACT_QA1010         DEDV              
DEDW D ACT101DM ACT_QA1010         DEDW              
DEDA D ACT101DM ACT_QA1010         DECA              
DECA D ACT101DM ACT_QA1010         DECA              
DECI D ACT101DM ACT_QA1010         DECI              
DECS D ACT101DM ACT_QA1010         DECS              
DEEG D ACT101DM ACT_QA1010         DEEG              
DECC D ACT101DM ACT_QA1010         DECC              
DEE5 D ACT101DM ACT_QA1010         DEEC              
DEEB D ACT101DM ACT_QA1010         DEEB              
DEFU D ACT101DM ACT_QA1010         DEFU              
DEFY D ACT101DM ACT_QA1010         DEFY              
DEEU D ACT101DM ACT_QA1010         DEEU              
DEGA D ACT101DM ACT_QA1010         DEGA        
```
Table 31 describes the columns on the Connections Table panel.

Table 31   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</td>
</tr>
<tr>
<td></td>
<td>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.
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.
Working with lists and searches

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 .................................................. 181
  Generating a combined list ...................................... 182
  Excluding objects from a combined list ....................... 183
Using SEARCH to generate lists based on object attributes 184
  Valid objects for searches ...................................... 185
  Generating a list by using the SEARCH command .......... 186
  Using host variables in a search ............................... 191
Using the Quick-Search feature .................................. 194
  Using saved search variables in a Quick-Search .......... 194
  Using a WHERE clause in a Quick-Search ................... 195
  Including Quick-Screens in BATCH jobs ..................... 196
Creating complex searches ....................................... 196
  Creating searches that do not contain a JOIN ............. 196
  Creating searches that contain a JOIN ....................... 197
Customizing object list displays .............................. 198
  Specifying a new order for displayed columns ............. 198
  Using the traditional list line format ...................... 200
  Sorting a list by one or more columns ...................... 201
  Locating string values ........................................ 202
  Counting items ................................................... 202
Describing list objects ........................................... 204
Generating JCL for a job in batch ............................... 209
  Using the BATCH command for a DB2 object list or a mixed list 210
  Using the BATCH command for a CATALOG MANAGER list or search 214
Generating, editing, and executing SQL ....................... 215
  Using Confirm SQL panels .................................... 216
  Working with the SQL_Table ................................ 217
  Applying SQL model statements ............................... 219
  Extended SQL processing ...................................... 222
Where to go from here ............................................ 225
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.

NOTE
For more information, view the Quick Course Understanding List Processing. You must have a BMC Support ID to view the Quick Course.

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 209.

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 32 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.
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).
Generating a mixed list

Figure 71  Table Space List panel to generate a mixed list

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

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 33 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.
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.
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).

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 34 lists the source objects for which you can perform searches.

Table 34  Valid source objects for searches (part 1 of 2)

<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>
<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>
</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 184.

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 34  Valid source objects for searches (part 2 of 2)

<table>
<thead>
<tr>
<th>Object type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<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>
<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.*
In the **Oper** column, type the operators that indicate the type of comparison that CATALOG MANAGER should use for the search.

Table 35 lists commonly used operators that are valid on a search panel.

### Table 35  Valid search operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
</table>
| =        | selects objects with values equal to the **Value** field  
This operator is the default for all attributes. |
| >        | selects objects with values greater than the **Value** field |
| <        | selects objects with values less than the **Value** field |
| LIKE     | selects objects with values that match the wildcard pattern entered in the **Value** field |
| ^=       | selects objects with values that are not equal to the **Value** field |
| <>       | selects objects with values that are not greater than the **Value** field |
| <=       | selects objects with values that are not less than the **Value** field |
| =B       | selects objects that have blank values for this attribute |
3 In the **Value** column, type the values for the fields that CATALOG MANAGER will compare in the search.

**NOTE**
If the value contains wildcard characters, such as % and _, you must use operators that mean LIKE or NOT LIKE. Otherwise, the wildcard characters are interpreted as literal characters when the search is processed.

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** **N (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 192.

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 12, “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.
Generating a list by using the SEARCH command

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.

F 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.

7 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

![Search Options panel](image)
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.

D Press Enter.

- A Search Variables List panel is displayed when multiple sets of variables match the owner or name values.
- The appropriate search panel is displayed when only one set of variables matches the name value.

E In the Command column, choose one of the following actions:

- Type DELETE (DEL) to delete a set of variables. The DELETE action deletes the line and displays the Search Variables List panel so that you can process another command.
- Type SELECT (S) to select a set of variables. The SELECT action displays the appropriate search panel. You can edit the search variables or apply them to a new search.

F Press Enter to process the line command.

8 Press Enter to perform the search, or press END to display the Primary Menu panel or list panel.

### 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 184.
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:
   \[
   \text{CREATOR LIKE :CREATOR AND NAME = :NAME}
   \]
   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).
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 36 describes the valid host variable values.

### Table 36 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>
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

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 a WHERE clause in a Quick-Search

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 214.

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 184.

   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 184.

   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.

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:

   ```sql
   . SYSIBM.SYSTABLES B
   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')
   ```
Customizing object list displays

CATALOG MANAGER offers you the ability to view the information on an object list display in the traditional list line format or to customize the display of information.

When you use the traditional list line format, CATALOG MANAGER displays all of the columns of information on an 80-character panel. You cannot specify which columns are displayed or the order in which those columns are displayed. By default, information is displayed in the traditional list line format.

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.

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. 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**

<table>
<thead>
<tr>
<th>Command ====&gt;</th>
<th>Scroll ====&gt;</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF-R---------</td>
<td>Column Order Specification</td>
<td>ROW 1 OF 47</td>
</tr>
<tr>
<td>Specify sequence numbers to place columns in desired order, then press END. Type SAVE command to save the selected order as the list default.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Order Colno Column Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TRADITIONAL LIST LINE FORMAT</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1 NAME</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2 CREATOR</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3 DBNAME</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4 DBID</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5 OBID</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6 PSID</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7 BPOOL</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8 PARTITIONS</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>9 LOCKRULE</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>10 PGSIZE</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>11 ERASERULE</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>12 STATUS</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>13 IMPLICIT</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>14 NTABLES</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>15 NACTIVE</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>16 DSETPASS</td>
<td></td>
</tr>
</tbody>
</table>

4 In the **Order** field, specify a new sequence number for each of the columns that you want to re-order.

5 For the traditional list line format column name, type **99** in the **Order** field.

6 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.
Using the traditional list line format

If you have not changed the order for the columns in a list and saved your changes, CATALOG MANAGER displays a list of objects in the traditional list line format. If you changed the order (see “Specifying a new order for displayed columns” on page 198), you can display the columns in the traditional list line format again by performing the following steps.

1 Generate a list of objects.

   For information, see “Generating lists in CATALOG MANAGER” on page 47.

2 On the Command line, type ORDER.

3 Press Enter.

4 Locate the traditional list line format column name.

5 In the Order column, type 1 for the traditional list line format.

6 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: 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).

Figure 79  Sort Specifications panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF-R----------</td>
<td>SORT SPECIFICATIONS</td>
<td>ROW 1 OF 46</td>
</tr>
<tr>
<td>Select columns to sort using digits 1-9. Specify D for descending order. Type SAVE command to save the specified sort as the default for the list.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQUENCE ASC/DESC NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>NAME</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>CREATOR</td>
</tr>
<tr>
<td>1</td>
<td>A</td>
<td>DBNAME</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>DBID</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>OBID</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>PSID</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>BPOOL</td>
</tr>
<tr>
<td>2</td>
<td>D</td>
<td>PARTITIONS</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>LOCKRULE</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>PGSIZE</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>ERASERULE</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>STATUS</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>IMPPLICIT</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>NTABLES</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>NACTIVE</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>DSETPASS</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>CLOSERULE</td>
</tr>
</tbody>
</table>
4 In the **SEQUENCE** column, specify a sort order for the columns. You can sort from 1 to 9 columns.

5 In the **ASC/DESC** column, type **A** or **D** for any of the columns that you want displayed in ascending or descending order.

6 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 549.

### 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.
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 97.)

---

**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.
Figure 82  Describe Table panel generated from DESCRIBE command

<table>
<thead>
<tr>
<th>Table: QZU.QZUT01_DEBS01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
</tr>
<tr>
<td>Table = QZU.QZUT01_DEBS01</td>
</tr>
<tr>
<td>FROM SYSIBM.SYSTABLES</td>
</tr>
<tr>
<td>Creator. . . . . QZU</td>
</tr>
<tr>
<td>Table Name . . . QZUT01_DEBS01</td>
</tr>
<tr>
<td>Database . . . QZUDEB</td>
</tr>
<tr>
<td>Tablespace . . . QZUS01EB</td>
</tr>
<tr>
<td>Editproc . . .</td>
</tr>
<tr>
<td>Audit. . . . .</td>
</tr>
<tr>
<td>Pctpages . . . . 75</td>
</tr>
<tr>
<td>Colcount . . . . 12</td>
</tr>
<tr>
<td>Record Length. . . 86</td>
</tr>
<tr>
<td>Key Columns. . . 5</td>
</tr>
<tr>
<td>Createdby. . . . RDABKH1</td>
</tr>
<tr>
<td>Remarks. . . .</td>
</tr>
<tr>
<td>Altereds. . . . -06-10.40.20.849236</td>
</tr>
<tr>
<td>Rba1 . . . . . X'006009FB9000'</td>
</tr>
<tr>
<td>Pctrowcomp . . . 70</td>
</tr>
<tr>
<td>Orphan . . . .</td>
</tr>
<tr>
<td>IBM reqd . . . . I</td>
</tr>
<tr>
<td>Viewdeps . . . . Cardf. . . 988013</td>
</tr>
<tr>
<td>Checkrid5b . . . X'4040404040'</td>
</tr>
<tr>
<td>Tbcreator. . . .</td>
</tr>
<tr>
<td>Npagesf. . . . . 14768</td>
</tr>
<tr>
<td>Avgrowlen. . . . 60</td>
</tr>
<tr>
<td>Tablestatus. . .</td>
</tr>
<tr>
<td>Version. . . . . 0</td>
</tr>
<tr>
<td>Split Rows . . .</td>
</tr>
<tr>
<td>Owner. . . . . . QZU</td>
</tr>
<tr>
<td>Ownertype. . . .</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>TABLE CONSTRAINTS</td>
</tr>
<tr>
<td>Constraint Name</td>
</tr>
<tr>
<td>Type Enforcing Index</td>
</tr>
<tr>
<td>Creator</td>
</tr>
<tr>
<td>Colcount</td>
</tr>
<tr>
<td>DATE</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>QZU.QZUX02_DEBS01T01</td>
</tr>
<tr>
<td>RDABKH1</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>TABLE COLUMNS</td>
</tr>
<tr>
<td>Num Column Name</td>
</tr>
<tr>
<td>Coltype</td>
</tr>
<tr>
<td>Length</td>
</tr>
<tr>
<td>Nl</td>
</tr>
<tr>
<td>Up</td>
</tr>
<tr>
<td>Pkey</td>
</tr>
<tr>
<td>Dfl</td>
</tr>
<tr>
<td>Bit</td>
</tr>
<tr>
<td>Length2</td>
</tr>
<tr>
<td>1 DATE</td>
</tr>
<tr>
<td>2 AUTHID</td>
</tr>
<tr>
<td>3 TBCREATOR</td>
</tr>
<tr>
<td>4 TBNAME</td>
</tr>
<tr>
<td>5 PLAN</td>
</tr>
<tr>
<td>6 LAST_MNT_DATE_TIME</td>
</tr>
<tr>
<td>7 NTRANS</td>
</tr>
<tr>
<td>8 INSERTS</td>
</tr>
<tr>
<td>9 UPDATES</td>
</tr>
<tr>
<td>10 DELETS</td>
</tr>
<tr>
<td>11 MIN_TIME</td>
</tr>
<tr>
<td>12 MAX_TIME</td>
</tr>
<tr>
<td>Row length - maximum 86 - minimum 86 including eight byte header</td>
</tr>
</tbody>
</table>

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 37.

**Table 37  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 37, 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>
<thead>
<tr>
<th>Command =&gt;</th>
<th>Table: QZU.QZUTO1_DEBS01</th>
<th>Line 1 of 72 Col 1 80</th>
<th>Scroll =&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table = QZU.QZUTO1_DEBS01</td>
<td>From SYSIBM.SYSTABLES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creator . . . QZU</td>
<td>Type . . . T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table Name . . . QZUTO1_DEBS01</td>
<td>Status . . . X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database . . . QZUDEB</td>
<td>Checkflag . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tablespace . . . QZUS01EB</td>
<td>CheckrId . . . X‘40404040’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Editproc . . .</td>
<td>ValIdproc . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audit . . .</td>
<td>Parents . . . 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pctpages . . . 75</td>
<td>Children . . . 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Colcount . . . 12</td>
<td>DBID . . . 1204</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Record Length . 86</td>
<td>OBID . . . 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Key Columns . . . 5</td>
<td>Key OBID . . . 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Createdby . . . RDABKH1</td>
<td>Label . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remarks . . .</td>
<td>Createdts . . . ..-06-10.40.20.849236</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alteredts . . . ..-06-10.40.20.849236</td>
<td>Datacapture . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rba1 . . . X’006009FB9000’</td>
<td>Rba2 . . . X’006009FB9000’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pctrowcomp . . . 70</td>
<td>Statstime . . . ..-07-09.12.32.710646</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orphan . . .</td>
<td>Clustertype . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBM reqd . . . I</td>
<td>Checks . . . 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Viewdeps . . .</td>
<td>Cardf . . . 988013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CheckrId5b . . . X‘4040404040’</td>
<td>Encoding scheme . E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tbcreator . . .</td>
<td>Tbcname . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Npagesf . . . 14768</td>
<td>Spacef . . . 79920</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avgrowlen . . . 60</td>
<td>Relcreated . . 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tablestatus . .</td>
<td>Number dep MQTs . 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Version . . . 0</td>
<td>Partkeycolnum . 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Split Rows . . .</td>
<td>Security Label .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owner . . . QZU</td>
<td>Append . . N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owntype . . .</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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>2</td>
<td>AUTHID</td>
<td>CHAR</td>
<td>8</td>
<td>2</td>
<td>TSSNOTES</td>
<td>AWDPONR</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>TBCREATOR</td>
<td>CHAR</td>
<td>8</td>
<td>3</td>
<td>UZU</td>
<td>QZU</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>TBNAME</td>
<td>CHAR</td>
<td>18</td>
<td>4</td>
<td>ZZ</td>
<td>AS&amp;P</td>
<td>204</td>
</tr>
<tr>
<td>5</td>
<td>PLAN</td>
<td>CHAR</td>
<td>8</td>
<td>5</td>
<td>WITCP03</td>
<td>$A486</td>
<td>295</td>
</tr>
<tr>
<td>6</td>
<td>LAST_MNT_DATE_TIME</td>
<td>TIMESTAMP</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>NTRANS</td>
<td>INTEGER</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Describing list objects**

---

**208 CATALOG MANAGER for DB2 User Guide**
**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 38).

**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 \texttt{command BATCH} or \texttt{DB2command}
\texttt{BATCH} in the \texttt{Cmd} column of the source object, and press \texttt{Enter}.

The variable \textit{command} is one of the following commands:

- \texttt{CASCADE}
- \texttt{CONNECT} (can only be issued from a location, or LO, list)
- \texttt{DCL}
- \texttt{DDL}
- \texttt{DES}
- \texttt{DESCRIBE}
- \texttt{DESTATISTICS}
- \texttt{HDDL}
- \texttt{HDESCRIBE}
- \texttt{HGRANT}

The variable \textit{DB2command} is one of the following commands:

- \texttt{DISPLAY}
- \texttt{START}
- \texttt{STOP}

CATALOG MANAGER displays the following message:

\texttt{BMC14651 Use BATCH command to generate JCL for all saved commands.}

3 In the \texttt{Cmd} column of the source object or on the Command line, type \texttt{BATCH} and press \texttt{Enter}.

The CATALOG MANAGER Batch Job panel is displayed (Figure 84).
4 In the **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 **Set JCL options** field, type **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 **Build Job** field, type **Y** to create the JCL and save it in the specified JCL data set.
8 In the **Edit Dataset** field, type **Y** to edit the JCL data set.

**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.

```
//SYSTSIN DD *
ISPSTART PGM(ACTBMAIN),PARM(O=DC91QEDK,S=DEDK,I=YES,V=DEDKCAT)
```

Versions 11.1 and later of CATALOG MANAGER are not dependent upon the ISPF interface. In these versions, CATALOG MANAGER replaces the SYSTSIN DD statement with the CATBATCH stepname.

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 30 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 38 describes the command syntax.

**Table 38 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><strong>DISPLAY objectType objectName</strong></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><strong>START objectType objectName</strong></td>
<td></td>
<td>DB, TS, FN&lt;sup&gt;ab&lt;/sup&gt;, PR&lt;sup&gt;ab&lt;/sup&gt;</td>
</tr>
<tr>
<td><strong>STOP objectType objectName</strong></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

   ```
   //SYSIN DD *
   DSN=RDACRJ2.DEDK.HDDL
   NO CATALOG MANAGER COMMANDS
   //** END OF JOBSTEP
   //** END OF JOB
   //** END OF JOBSTEP
   //*** Bottom of Data ***
   ```

5. Type over the message with one or more of the following commands (each on a separate line):
   - DOPTS
   - DSNZPARM
   - 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.

- **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 89).

Figure 86  Confirm SQL panel for CREATE table
Working with the SQL_Table

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.

Figure 87  SQL_Table List panel

```
DEFF-R -----------------------  SQL Table List  -------- Row 183 to 196 of 207
Command ===>                                                  Scroll ===> CSR
CMD will show commands for this list. Type command and press Enter         02
Subcommands are: ANALYZE, CUT, DELETE, EDIT, EXECUTE, PASTE, RENAME, TBBROWSE, TEDIT, 2WL. ANALYZE may be used with SELECT, INSERT, DELETE and UPDATE SQL.
SQL NAME LIKE: *.*
Cmd   Owner    Name               Text
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v---
RDAMSL   PBRTSTB            CREATE TABLESPACE MSLTSPBR IN MSLTMPDB LOGGE
RDAPKM   20100317_111620    DSN SYSTEM(DEFF) DCLGEN TABLE(SYSSTABLESPACES
RDAPXB   CREATE_TB          CREATE TABLE PXB.CURRENTQUOTE ( SYMBOL CHAR(5)
RDAPXB   CX                 CREATE TRUSTED CONTEXT QCH_TRUSTD_CONTEXT BA
RDAPXB   DROP_INCLUDE       CREATE UNIQUE INDEX TAD.TAD_TBL01_IX ON TAD.
RDAPXB   DTS                CREATE DISTINCT TYPE QCH.QCH_IDENT_1 AS INTEGER
RDAPXB   FN                 CREATE FUNCTION QCH.QCH_FN_EXT_TYPE_T ( LAST
RDAPXB   HDDL               -- HDDL OBJECTS PRODUCED FOR: TS TB VW IX SY
RDAPXB   PR                 CREATE PROCEDURE QCH.QCHCT007 ( IN INAME VAR
RDAPXB   SYSTEM_DBA         GRANT DBADM WITHOUT ACCESSCTRL WITHOUT DATAA
RDAPXB   TEST               -SQLP 000100 CREATE PROCEDURE TAD.TAD_NATIVE
RDAPXB  TEST-109           -- HDDL OBJECTS PRODUCED FOR: TS TB VW IX SY
RDAPXB   TRUNCATE_TEST      -- HDDL OBJECTS PRODUCED FOR: TS TB VW IX SY
RDAPXB2  EXECUTION          -- HDDL OBJECTS PRODUCED FOR: TS TB VW IX SY
```
Copy 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 217.

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 76.

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.
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).
Applying SQL model statements

7 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

<table>
<thead>
<tr>
<th>Command</th>
<th>Row 1 to 15 of 235</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actions</td>
<td></td>
</tr>
<tr>
<td>Edit options...</td>
<td>N (Y/N)</td>
</tr>
<tr>
<td>Edit SQL ......</td>
<td>N (Y/N)</td>
</tr>
<tr>
<td>Save SQL ......</td>
<td>N (A/R/Y/N A-append, R-replace, Y-append)</td>
</tr>
<tr>
<td>Execute ......</td>
<td>N (Y/N)</td>
</tr>
<tr>
<td>Options</td>
<td>Name of saved SQL : 20110209_325680</td>
</tr>
<tr>
<td></td>
<td>SQL</td>
</tr>
<tr>
<td></td>
<td>ALTER TABLE QZU . QZUT01_DCI08S01</td>
</tr>
<tr>
<td></td>
<td>ALTER TABLE QZU . QZUT02_DCI08S01</td>
</tr>
<tr>
<td></td>
<td>ALTER TABLE QZU . QZUT03_DCI08S01</td>
</tr>
</tbody>
</table>

8 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.
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).

Figure 90  Confirm SQL panel for extended SQL processing

![Confirm SQL panel](image)

3 In the Execute SQL field, type Y.

4 Press Enter.

The SQL Host Variables List panel is displayed (Figure 91).
In the Substitute Value column of the **DATE** row, type the value for the **DATE** host variable.

In the Substitute Value column of the **FUNCTION** row, type the value for the **FUNCTION** host variable.

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.
Where to go from here
Browsing and editing data

This chapter contains the following topics:

Overview ................................................................. 227
Browsing table data ....................................................... 228
  Invoking the data browsing function .............................. 228
  Setting options for browsing data ................................. 228
  Browsing data ........................................................ 233
  Browsing data in LOB columns .................................... 234
Editing table data ......................................................... 235
  Invoking the data editing function ................................. 235
  Setting options for editing data ..................................... 235
  Editing data .......................................................... 238
Copying table data ....................................................... 241
  Using the COPY command .......................................... 241
  Using the Copy Table Rows option ............................... 245
Where to go from here .................................................. 249

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 H, “Commands.”

NOTE
For more information, view the Quick Course Using the Table Editor. You must have a BMC Support ID to view the Quick Course.

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 39 lists the available methods for invoking the data browsing function.

<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 SELECT SQL statement</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 228.

   The Browse DB2 Table Options panel is displayed (Figure 92).
In the Current SQLID field, type a different SQLID.

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.

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 40 on the Select Statement Specification panel. By default, all columns are designated as “selected.”

**Table 40  SELECT statement specifications**

<table>
<thead>
<tr>
<th>Specification</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>remove a column from the SELECT statement&lt;sup&gt;a&lt;/sup&gt;</td>
<td>delete the S designator beside the column name</td>
</tr>
<tr>
<td>specify a sequence of columns&lt;sup&gt;a&lt;/sup&gt;</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 35 on page 187)</td>
</tr>
<tr>
<td>specify values for the search operation&lt;sup&gt;b&lt;/sup&gt;</td>
<td>type the values in the Value field</td>
</tr>
</tbody>
</table>

<sup>a</sup> You can use the S designator for some columns and a sequence number for other columns of the same table.

<sup>b</sup> 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 41 to clear changes that you have made.

**Table 41  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 Select statement name 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.
9 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.

10 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**

<table>
<thead>
<tr>
<th>Option</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Headings</td>
<td>1 - Names, 2 - Labels, 3 - Numbers</td>
</tr>
<tr>
<td>Date and Time defaults</td>
<td>1 - 1900-01-01-00.00.00.000000, 2 - Current date and time, 3 - Blank</td>
</tr>
<tr>
<td>Display update SQL</td>
<td>N - Display SQL used to update the table.</td>
</tr>
<tr>
<td>Confirm before update</td>
<td>N - Display confirm panel before saving changes.</td>
</tr>
<tr>
<td>Display Statistics</td>
<td>Y - Display Table Editing Statistics.</td>
</tr>
<tr>
<td>Browse with UR</td>
<td>Y - Browse DB2 rows using WITH UR in select.</td>
</tr>
<tr>
<td>CAPS ON</td>
<td>N - Uppercase input and modified values.</td>
</tr>
<tr>
<td>Default SQL Owner</td>
<td>S - Default owner for SQL unqualified sql table member names when saving or retrieving SQL, T - TSO ID, S - SQL ID</td>
</tr>
<tr>
<td>Memory Allocation Limit</td>
<td>0 - Meg. Maximum memory to allocate for DB2 rows, in megabytes, 0 = No limit.</td>
</tr>
<tr>
<td>Left Justify Numerics</td>
<td>N - Numeric fields in rowview mode will be displayed left justified.</td>
</tr>
<tr>
<td>Display Decimal Point</td>
<td>N - Include a decimal point in the value of columns with type DECIMAL(n,0)</td>
</tr>
<tr>
<td>Clear Editor Users Table</td>
<td>Y/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.</td>
</tr>
</tbody>
</table>
Table 42 shows the options that you can specify on the Edit and Browse Options panel.

### Table 42  Additional edit and browse options

<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><em>(editing data)</em> 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</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</td>
</tr>
<tr>
<td></td>
<td>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.EDITOR_USERS table (see “CATALOG MANAGER tables” on page 157)</td>
</tr>
<tr>
<td></td>
<td>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 228.

2. Set your options for browsing data. For more information, see “Setting options for browsing data” on page 228.
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 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 228.
3 In the Browse DB2 Table Options panel, specify your options.
   For more information, see “Setting options for browsing data” on page 228.
   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.
C Press Enter.

The data is displayed in an ISPF Browse panel.

D 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 43 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 <em>owner.tablename</em></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</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 235.

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
Follow the steps in “Setting options for browsing data” on page 228.

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  237

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 238.

### NOTE

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.

<table>
<thead>
<tr>
<th>To specify this locking option</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>row lock</td>
<td>R</td>
</tr>
</tbody>
</table>
| 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. |
| no lock                         | N    |
| 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. |
| - 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. |

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 43).
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 System User Guide, the BMC Products and Solutions for DB2 Configuration Guide, and the IBM documentation.

Editing data

1. Invoke the data editing function. For more information, see “Invoking the data editing function” on page 235.

2. Set the options for controlling the display and the SQL processing. For more information, see “Setting options for editing data” on page 235.
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.
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).

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 235.

2 On the Edit DB2 Table Options panel, set option values for editing data. For information, see “Setting options for editing data” on page 235.
Using the COPY command

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 ===>

Specify 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**

8 Customize the SELECT statement by modifying the selected columns and values on the panel. Press HELP to display example specifications.
Using the COPY command

**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 191.

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

<table>
<thead>
<tr>
<th>Command ====&gt;</th>
<th>Scroll ====&gt;</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZU.QZUT01_DCI19505 (301/2288)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*INS 1818599</td>
<td>COLUMN_1</td>
<td>COLUMN_2</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>AABNO000467</td>
</tr>
<tr>
<td>*INS 1818629</td>
<td>7364</td>
<td>AABNO000478</td>
</tr>
<tr>
<td>*INS 1818687</td>
<td>26700</td>
<td>AABNO000495</td>
</tr>
<tr>
<td>*INS 1818706</td>
<td>-20104</td>
<td>AABNO000501</td>
</tr>
<tr>
<td>*INS 1818722</td>
<td>27713</td>
<td>AABNO000506</td>
</tr>
<tr>
<td>*INS 1818733</td>
<td>-14913</td>
<td>AABNO000510</td>
</tr>
<tr>
<td>*INS 1818754</td>
<td>7200</td>
<td>AABNO000517</td>
</tr>
<tr>
<td>*INS 1818758</td>
<td>16000</td>
<td>AABNO000519</td>
</tr>
<tr>
<td>*INS 1818810</td>
<td>3300</td>
<td>AABNO000536</td>
</tr>
<tr>
<td>*INS 1818836</td>
<td>6800</td>
<td>AABNO000543</td>
</tr>
<tr>
<td>*INS 1818895</td>
<td>0</td>
<td>AABNO000561</td>
</tr>
<tr>
<td>*INS 1818902</td>
<td>0</td>
<td>AABNO000562</td>
</tr>
<tr>
<td>*INS 1818932</td>
<td>4275</td>
<td>AABNO000573</td>
</tr>
<tr>
<td>*INS 1818944</td>
<td>0</td>
<td>AABNO000577</td>
</tr>
<tr>
<td>*INS 1818954</td>
<td>5985</td>
<td>AABNO000578</td>
</tr>
<tr>
<td>*INS 1447973</td>
<td>18000</td>
<td>AABNO000585</td>
</tr>
<tr>
<td>*INS 1819036</td>
<td>0</td>
<td>AABNO000600</td>
</tr>
<tr>
<td>*INS 1448522</td>
<td>0</td>
<td>AABNO000603</td>
</tr>
<tr>
<td>*INS 1448585</td>
<td>-31536</td>
<td>AABNO000612</td>
</tr>
</tbody>
</table>

**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 235.

2 On the Edit DB2 Table Options panel, set option values for editing data. For information, see “Setting options for editing data” on page 235.

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).
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.
Using the Copy Table Rows option

Chapter 6 Browsing and editing data 247

Figure 102 Select Statement Specification panel

![Select Statement Specification panel]

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 191.

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).
Using the Copy Table Rows option

Figure 103 ISPF panel for editing INSERT statement

```
EDIT RDACRJ.BMCCAT.WORK                         Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
****** ******************************************************************* Top of Data ****************************
000001 INSERT INTO QZU.QZUT01_DCISS02 (  
000002 COLUMN_1,  
000003 COLUMN_2,  
000004 COLUMN_3,  
000005 COLUMN_4,  
000006 COLUMN_5,  
000007 COLUMN_6,  
000008 COLUMN_7,  
000009 COLUMN_8,  
000010 COLUMN_9,  
000011 COLUMN_10,  
000012 COLUMN_11,  
000013 COLUMN_12,  
000014 COLUMN_13,  
000015 COLUMN_14,  
000016 COLUMN_15,  
000017 COLUMN_16,  
000018 COLUMN_17,  
000019 COLUMN_18,
```

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>
Where to go from here
Creating objects

This chapter contains the following topics:

Overview ................................................................. 251
Before you begin ....................................................... 251
Estimating space requirements based on user-specified values .................................. 252
  Estimating space requirements for a table space ........................................ 253
  Estimating space requirements for an index ......................................... 256
Using an existing object as a model to create objects ........................................... 257
  Considerations for creating objects .................................................. 257
  Example of creating a table .................................................... 258
Generating DDL to create objects .................................................. 271
Where to go from here .................................................... 274

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.
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 94.

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 252.

### Estimating space requirements based on user-specified values

Standard space-estimation tools use IBM default DDL values for object attributes when calculating estimates. In contrast, the Simple Space Estimation (SSE) feature allows you to replace those defaults with values that are specific to your objects. DASD MANAGER PLUS, CATALOG MANAGER, and CHANGE MANAGER support this feature.

**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.

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 253
- “To estimate space requirements for a partitioned table space” on page 254
- “To estimate space requirements for a partition-by-growth table space” on page 255

**NOTE**

Space estimation is not available for XML objects, LOB objects, or 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**.
Estimating space requirements for a table space

**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 253) 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 253) as needed to correspond to your table space.

---

**NOTE**
For more information about a specific field, press **F1**.
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 253) 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.
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, LOB objects, or 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 253) 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.
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.
Figure 105  Create/Alter Table panel

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).
Example of creating a table

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.
3 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 Opts** 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).
Figure 109  Additional column options panel

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**

![Column Identity Information panel](image)

- **Column name**: COLUMN_1
- **Generated**: Always, By default
- **Start/Restart with**
- **Increment by**: 1
- **MinValue**
- **MaxValue**
- **Cache**: Y
- **Cache Amount**: 20
- **Cycle**: N
- **Order**: N

**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 Constraints panel](image)

- **Table Creator**: QZU
- **Table Name**: QZUT00_DSC30S28
- **Edit Unique/Primary**: N
- **Edit Foreign Key**: N
- **Edit Check**: N
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.

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.

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.

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).
Figure 113  Select Generate Text panel

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.

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 257), 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 44 describes the different DDL commands.

<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) DDL</td>
<td>generates DDL to create the objects for which the command was entered</td>
<td>The settings for the fields on the following options panels affect the DDL:</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>(from the Command line) DDL objectCode ownerName.objectName</td>
<td>for multiple objects, generates individual DDL streams</td>
<td>— General Options: <strong>Decimal point</strong> and <strong>SQL string delimiter</strong> — SQL and Confirm Options: all fields — SQL Select: all fields — Switches: <strong>Define No</strong></td>
<td></td>
</tr>
</tbody>
</table>
Generating DDL to create objects

Table 44  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>
<tbody>
<tr>
<td>(from an object list) HDDL</td>
<td>generates DDL to create the objects for which the command was entered and for the dependent objects</td>
<td>- 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: --COMMENTED IMPLICIT</td>
<td>DB MQT NP PR TB TS VW</td>
</tr>
<tr>
<td>(from the Command line_) HDDL objectCode ownerName.objectName</td>
<td>for multiple objects, generates individual DDL streams for each object and its dependents</td>
<td>If the indexes were created explicitly, uncomment the DDL.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>enables you to include GRANT authorizations in the SQL</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>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 209.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 44 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><strong>HDDL (continued)</strong></td>
<td>The settings for the fields on the following options panels affect the HDDL:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>— Object Use Options: <strong>Include in HDDL and HDESCRIBE</strong> and <strong>Include in HDDL commit counts</strong></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: <strong>Define No</strong>, <strong>Cmp &gt; 32k</strong>, <strong>HDDL Auths</strong>, and <strong>Build SQLID before GRANT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If you specify N for the <strong>HDDL Auths</strong> 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 11, “Managing authorizations.”</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MDDL</strong> (from an object list)</td>
<td>generates DDL to create the objects for which the command was entered</td>
<td></td>
<td>AL CX DB DT FN IM IX MK MQT NP PM PR SE SG SY TB TR TS VW XT</td>
</tr>
<tr>
<td><strong>MDDL objectCode ownerName.objectName</strong></td>
<td>applies to only a single object type</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>for multiple objects of the same object type, generates one DDL stream for all of the objects</td>
<td></td>
<td></td>
</tr>
<tr>
<td></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>
<td></td>
</tr>
<tr>
<td></td>
<td>The settings for the fields on the following options panels affect the MDDL:</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>The BATCH keyword is not valid for the MDDL command.</td>
<td></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 ................................................................. 275
Considerations for dropping objects ................................ 275
  Considerations for dropping a table space ...................... 276
  Considerations for dropping pending changes ............... 276
  Considerations for dropping a table ............................ 276
  Considerations for dropping an index ......................... 277
Simulating a drop ...................................................... 278
Dropping an object .................................................... 278
Recovering an object and its data ................................ 283
Where to go from here .............................................. 288

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 94.

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 278.

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.
Dropping an object

Chapter 8 Dropping and recovering objects

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**

<table>
<thead>
<tr>
<th>Dependent Objects for TABLESPACE: QZUDAC.QZUS01AC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
</tr>
<tr>
<td>.TB</td>
</tr>
<tr>
<td>..IX</td>
</tr>
<tr>
<td>..IX</td>
</tr>
<tr>
<td>..IX</td>
</tr>
<tr>
<td>..IX</td>
</tr>
<tr>
<td>..IX</td>
</tr>
<tr>
<td>..IX</td>
</tr>
<tr>
<td>..IX</td>
</tr>
</tbody>
</table>

Déf -R Drop Dependency List Row 1 to 8 of 8 Command ===> Scroll ===> PAGE

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.

**M** *(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 278), 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>DEFF-R</th>
<th>----------------------</th>
<th>TABLESPACE LIST</th>
<th>----------------------</th>
<th>DROP RC= 000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>CMD will show commands for this list. Type command and press ENTER</td>
<td>Tablespace</td>
<td>Owner</td>
<td>Segsz</td>
</tr>
</tbody>
</table>
| Lists: AL BMCUHIST CA CL CO DB DS FK IC IM IS IX LK MOT MX NP OS PA PDD PG PL LIKE QZU%

<table>
<thead>
<tr>
<th>.Cmd</th>
<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>----</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>----</td>
<td>----</td>
<td>--------</td>
<td>------</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td><strong>drop AC.QZUS01AC</strong></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>
<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>
<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>
<td></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>
<td></td>
</tr>
<tr>
<td>QZUDAC.QZUS05AC</td>
<td>ASUQA</td>
<td>0</td>
<td>BP0</td>
<td>4</td>
<td>1</td>
<td>24K</td>
<td>A</td>
<td>E</td>
<td></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>
<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>
<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>
<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>
<td></td>
</tr>
<tr>
<td>QZUD1.QZUS01A1</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>
<td></td>
</tr>
<tr>
<td>QZUD1.QZUS02A1</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>
<td></td>
</tr>
<tr>
<td>QZUD1.QZUS03A1</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>
<td></td>
</tr>
<tr>
<td>QZUD1.QZUS04A1</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>
<td></td>
</tr>
<tr>
<td>QZUD1.QZUS05A1</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>
<td></td>
</tr>
<tr>
<td>QZUB1.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>
<td></td>
</tr>
<tr>
<td>QZUB1.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>
<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 278.

**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>Date</th>
<th>Time</th>
<th>AuthId</th>
<th>Product</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011-02-01</td>
<td>14.43</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>DBXNAUT.SBXNCOL</td>
</tr>
<tr>
<td>2011-01-19</td>
<td>16.25</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>PLBDBA1.PLBLALT02</td>
</tr>
<tr>
<td>2011-01-19</td>
<td>14.09</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>PLBDBA1.PLBLALTER</td>
</tr>
<tr>
<td>2011-01-17</td>
<td>14.16</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>PXBQXPB..HASHPXB1</td>
</tr>
<tr>
<td>2011-01-10</td>
<td>16.33</td>
<td>SKH2</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-10</td>
<td>16.13</td>
<td>SKH2</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-10</td>
<td>16.01</td>
<td>SKH2</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-10</td>
<td>15.39</td>
<td>SKH2</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-10</td>
<td>15.27</td>
<td>SKH2</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-10</td>
<td>15.25</td>
<td>SKH2</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-08</td>
<td>12.58</td>
<td>RDAMSX</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-08</td>
<td>11.25</td>
<td>RDAMSX</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-07</td>
<td>18.21</td>
<td>RDAMSX</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-07</td>
<td>17.44</td>
<td>RDAMSX</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-07</td>
<td>16.58</td>
<td>SKH</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</td>
</tr>
<tr>
<td>2011-01-07</td>
<td>16.44</td>
<td>SKH</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5..MXSS01G5</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 271).

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.
CATALOG MANAGER automatically excludes the statements that are shown in Table 45. These statements are displayed on the Recovery Statements panel, but are marked with X for exclusion from processing.

Table 45  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.
Recovering an object and its data

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.
To restart the table space, type \text{START} in the \text{Cmd} column of the \text{REC DATA} text line. Press \text{Enter}.

The DB2 Commands panel is displayed.

Press \text{Enter} to execute the command.

CATALOG MANAGER sends the command to DB2.

Press END.

The Drop Recovery List is displayed.

Type \text{S} in the \text{Cmd} column beside the table space to be recovered, and then press \text{Enter}.

The Recovery Statements panel is displayed.

To recover indexes, complete the following steps:

\begin{enumerate}
\item Remove the \text{X} notations from any indexes that you marked for exclusion.
\item In the \text{Execute recovery statements now} field, type \text{Y}.
\item Press \text{Enter}.
\end{enumerate}

The table space and data recovery are complete.

\section*{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 ................................................................. 289
Available utilities ..................................................... 290
   BMC products ...................................................... 290
   IBM DB2 utilities .................................................. 291
   Issuing utility commands .................................... 292
Utility profiles .......................................................... 292
   Creating a user utility profile data set ..................... 293
   Creating a utility profile for a single utility .......... 295
   Creating a utility profile for multiple utilities .... 300
   Creating a utility profile from an existing profile .. 302
   Editing a utility profile ....................................... 304
   Working with the last-used utility profile .......... 305
Switching utility profiles ............................................ 306
TEMPLATE and LISTDEF control statements .................. 306
   Available utilities for TEMPLATE and LISTDEF statements .. 307
   Incorporating TEMPLATE and LISTDEF statements .. 307
Where to go from here .................................................. 310

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 46. For information about issuing commands in CATALOG MANAGER, see “Listing and executing commands” on page 56.

Table 46  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></td>
</tr>
<tr>
<td>BMCCHECK IX</td>
<td></td>
</tr>
<tr>
<td>BMCCHTS</td>
<td></td>
</tr>
<tr>
<td>BMCCOPY</td>
<td>COPY PLUS for DB2</td>
</tr>
<tr>
<td>BMCCOPY INDEX</td>
<td></td>
</tr>
<tr>
<td>BMCCOPY IX</td>
<td></td>
</tr>
<tr>
<td>BMCCREBUILD</td>
<td>RECOVER PLUS for DB2</td>
</tr>
<tr>
<td>BMCCREBUILD INDEX</td>
<td></td>
</tr>
<tr>
<td>BMCCREBUILD IX</td>
<td></td>
</tr>
<tr>
<td>BMCRECOVER</td>
<td></td>
</tr>
<tr>
<td>BMCRECOVER IX</td>
<td></td>
</tr>
<tr>
<td>BMCRECOVER INDEX</td>
<td></td>
</tr>
<tr>
<td>BMCRECOVER IX</td>
<td></td>
</tr>
<tr>
<td>BMCREORG</td>
<td>REORG PLUS for DB2</td>
</tr>
<tr>
<td>BMCREORG INDEX</td>
<td></td>
</tr>
<tr>
<td>BMCREORG IX</td>
<td></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></td>
</tr>
<tr>
<td>BMCSTATS IX</td>
<td></td>
</tr>
<tr>
<td>BMCUNLOAD</td>
<td>UNLOAD PLUS for DB2</td>
</tr>
</tbody>
</table>
CATALOG MANAGER Help panels and Appendix F, “JCL Generation product options,” 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 47. For information about issuing commands in CATALOG MANAGER, see “Listing and executing commands” on page 56.

**Table 47 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>CHKD</td>
<td></td>
</tr>
<tr>
<td>CHECK INDEX</td>
<td>CHECK INDEX</td>
</tr>
<tr>
<td>CHECK IX</td>
<td></td>
</tr>
<tr>
<td>CHKI</td>
<td></td>
</tr>
<tr>
<td>COPY</td>
<td>COPY</td>
</tr>
<tr>
<td>COPY INDEX</td>
<td></td>
</tr>
<tr>
<td>COPY IX</td>
<td></td>
</tr>
<tr>
<td>COPYTOCOPY</td>
<td>COPYTOCOPY</td>
</tr>
<tr>
<td>COPYTOCOPY INDEX</td>
<td></td>
</tr>
<tr>
<td>COPYTOCOPY IX</td>
<td></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></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></td>
</tr>
<tr>
<td>REBUILD IX</td>
<td></td>
</tr>
<tr>
<td>RECOVER</td>
<td>RECOVER</td>
</tr>
<tr>
<td>RECOVER INDEX</td>
<td></td>
</tr>
<tr>
<td>RECOVER IX</td>
<td></td>
</tr>
<tr>
<td>REORG</td>
<td>REORG TABLESPACE</td>
</tr>
<tr>
<td>REORG INDEX</td>
<td></td>
</tr>
<tr>
<td>REORG IX</td>
<td></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.

NOTE

For more information, view the Quick Course User Utility Profiles. You must have a BMC Support ID to view the Quick Course.

Table 47 Commands to invoke IBM DB2 utilities (Part 2 of 2)

<table>
<thead>
<tr>
<th>CATALOG MANAGER command</th>
<th>IBM DB2 utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPORT</td>
<td>REPORT</td>
</tr>
<tr>
<td>REPORT INDEX</td>
<td></td>
</tr>
<tr>
<td>REPORT IX</td>
<td></td>
</tr>
<tr>
<td>RUNSTATS</td>
<td>RUNSTATS</td>
</tr>
<tr>
<td>RUNSTATS INDEX</td>
<td></td>
</tr>
<tr>
<td>RUNSTATS IX</td>
<td></td>
</tr>
<tr>
<td>STOSPACE</td>
<td>STOSPACE</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>UNLOAD</td>
</tr>
</tbody>
</table>

CATALOG MANAGER Help panels and Appendix H, “Commands,” provide more information about using these commands. For information about the IBM DB2 utilities, see the IBM documentation.
Creating a user utility profile data set

This procedure describes how to create a utility profile data set.

NOTE
Do not create or edit a PDS member for a utility data set outside of CATALOG MANAGER. These members are used by CATALOG MANAGER internally.

To specify a new data set

1. From the Primary Menu panel, on the Command line, type OPTIONS (OPT).
2. Press Enter.

The Options panel is displayed.
In the Edit Dataset names field, type Y.

The Datasets panel is displayed.

In the User Utilities Profile Dsn field, type the names of a data set and a member (for example, RDAABC.BMCCAT.UTILPROF(UTILPROF)).

**NOTE**
You cannot save your utility profiles in the same member in which you save your user options.

Press END to return to the Options panel.

**To allocate the data set**

1. Generate a list of table spaces.

   For more information, see “Generating lists in CATALOG MANAGER” on page 47.

2. On the list panel, type COPY in the Cmd column beside a table space.

3. Press Enter.

   The Utility List panel is displayed, with the name of the utility profile data set displayed in the User Profile Dsn field.

4. In the Cmd column, type ED.

5. Press Enter.

   The Copy utility panel is displayed.

6. In the Utility Id field, type the name of a utility ID.

7. On the Command line, type PROFILE SAVEAS DEFAULT.

   The Allocate Data Set panel is displayed.

8. In the Allocate data set with the following values field, type Y.


   The product displays a message that indicates that the profile name has been saved in the member and data set name that you specified. You can create utility profiles in the new member and 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 46 and Table 47. 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.

4. Press Enter.

The Utility List panel is displayed. Figure 121 shows the Utility List panel for the REORG PLUS utility.
Creating a utility profile for a single 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 87.

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.
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 48 shows valid variables for the Utility ID.

Table 48  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</td>
<td>the database of the first table space that is selected</td>
</tr>
<tr>
<td>&amp;ACTUTS</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>

\* &ACTUTS 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 543.

   **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

Figure 122 REORG PLUS utility panel

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility Id</td>
<td>Reorg Plus: QZUD34.QZUS0134</td>
</tr>
<tr>
<td>Part list</td>
<td>0-4096 new parts to add</td>
</tr>
<tr>
<td>Share Level</td>
<td>Enter ? for list</td>
</tr>
<tr>
<td>Analyze</td>
<td>Y/N/P/O Y-Yes, N-No, P-Pause, O-Only</td>
</tr>
<tr>
<td>Analyze cardinality</td>
<td>S/C/H/B S-Sample, C-Scan, H-Hurba, B-BMCSTATS</td>
</tr>
<tr>
<td>Unload</td>
<td>C/P/R C-Continue, P-Pause, R-Reload</td>
</tr>
<tr>
<td>Keepdictionary</td>
<td>Y/N Keep compression dictionary</td>
</tr>
<tr>
<td>Message level</td>
<td>O/1 O-Minimal, 1-Diagnostic msgs</td>
</tr>
<tr>
<td>Shrlevel change</td>
<td>N Edit Shrlevel change options</td>
</tr>
<tr>
<td>Sort options</td>
<td>N Edit Sort options</td>
</tr>
<tr>
<td>On failure options</td>
<td>N Edit On Message and/or On Failure</td>
</tr>
<tr>
<td>Statistics options</td>
<td>N Edit Statistics options</td>
</tr>
<tr>
<td>Copy options</td>
<td>N Edit Copy options</td>
</tr>
<tr>
<td>DD options</td>
<td>N Edit DD options</td>
</tr>
<tr>
<td>Limit options</td>
<td>N Edit Limit options</td>
</tr>
<tr>
<td>Sel/Del/Upd options</td>
<td>N Edit Select/Delete/Update options</td>
</tr>
<tr>
<td>Additional options</td>
<td>N Edit Additional options</td>
</tr>
</tbody>
</table>

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</th>
<th>1 to 6 of 6</th>
<th>Scroll ===&gt;</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>JCL Dataset.</td>
<td>'RDACRJ.BMCCAT.JCL(REORPLUS)'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Profile Dsn.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set JCL options.</td>
<td>N Y/N - Change options for creating JCL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Build Job.</td>
<td>Y Y/N - Create JCL, save in dataset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Dataset</td>
<td>Y Y/N - Edit JCL dataset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Submit</td>
<td>N Y/N - Submit JCL dataset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Utility ID</td>
<td>RBLDPLUS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<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 49.
Creating a utility profile for multiple utilities

| Table 49 Status column values for the Utility List panel |
|---------------------------------|--------------------------------------------------|
| Value  | Description                                                                 |
| blank  | You did not enter any commands for this statement.                           |
| Edited | You entered the ED command for this statement.                               |
| Init   | When the Status value of the statement was blank, the product generated JCL for this statement. |
| Profil | The product initialized the options from a utility profile.                 |
| Propgt | The product propagated the options from the previous object.                |
| Single | The product combined two or more objects into one single statement.          |

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** (**UTIL**) 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>Command ===&gt;</th>
<th>Utility Selections for Table space</th>
<th>Scroll ===&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select the utility by typing a number next to the utility. Valid values are 0 through 8. If you want to use a utility more than once, separate the numbers with a comma. For example, to use COPY, LOAD, and COPY, type 1,3 for COPY and 2 for LOAD. To save, type PROFILE SAVEAS profileID.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC Utilities</td>
<td>BMCSTATS . . . . .</td>
<td>COPY PLUS . . . .</td>
</tr>
<tr>
<td></td>
<td>CHECK PLUS (Data). .</td>
<td>LOADPLUS . . . .</td>
</tr>
<tr>
<td></td>
<td>CHECK PLUS (TS). .</td>
<td></td>
</tr>
<tr>
<td>IBM Utilities</td>
<td>CHECK DATA . . . .</td>
<td>LOAD . . . .</td>
</tr>
<tr>
<td></td>
<td>CHECK INDEX. . .</td>
<td>MERGECOPY . . .</td>
</tr>
<tr>
<td></td>
<td>COPY . . . .</td>
<td>MODIFY RECOVERY . .</td>
</tr>
<tr>
<td></td>
<td>COPYTOCOPY . . .</td>
<td>MODIFY STATISTICS.</td>
</tr>
<tr>
<td></td>
<td>DSN1COMP . . .</td>
<td>QUIESCE. . .</td>
</tr>
<tr>
<td></td>
<td>DSN1COPY . . .</td>
<td>REBUILD INDEX.</td>
</tr>
<tr>
<td></td>
<td>EXEC SQL . . .</td>
<td>RECOVER. . .</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 295.

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 46 on page 290 and Table 47 on page 291.

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

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 46 on page 290 and Table 47 on page 291.

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 94.

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.

Selective 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 option 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 130.
Available utilities for TEMPLATE and LISTDEF statements

Table 50 lists the IBM utilities for which TEMPLATE and LISTDEF statements are valid.

Table 50  Valid IBM utilities for TEMPLATE and LISTDEF statements

<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 47 on page 291.

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.
Incorporating TEMPLATE and LISTDEF statements

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

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).
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.
Where to go from here

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 11, “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 12, “Customizing CATALOG MANAGER command access”</td>
</tr>
<tr>
<td>Browse and purge logs that CATALOG MANAGER maintains.</td>
<td>Chapter 13, “Maintaining logs”</td>
</tr>
</tbody>
</table>
Issuing DB2 commands

This chapter presents the following topics:

Overview ................................................................. 311
Methods for issuing commands ................................... 311
  Using command prompts ........................................... 312
  Using model commands ............................................ 315
  Specifying command syntax ..................................... 316
DB2 command profiles ................................................ 318
Where to go from here ................................................. 320

Overview

This chapter describes how to issue DB2 commands against objects in a list.

**NOTE**

For more information, view the Quick Course Using DB2 Commands. You must have a BMC Support ID to view the Quick Course.

For a list of DB2 commands, see Appendix H, “Commands.”

Methods for issuing commands

You can issue DB2 commands by using the following methods:

- *command prompts*, which guide you through the process of specifying keywords
- *model commands*, which provide a list of the commonly-used commands
- *command syntax*, which enables you to directly specify the command and keywords
Using command prompts

When you create a DB2 command, CATALOG MANAGER provides you with a user-friendly interface that has predefined input fields. These input fields correspond to the keywords for a DB2 command. The interface also provides an optional field in which you can specify additional keywords for the command.

The following procedure describes how to use the command prompt interface.

1. In the Action field on the Primary Menu panel, select DB2 Commands and press Enter.

   CATALOG MANAGER displays the DB2 Commands List panel (Figure 128).

   **Figure 128  DB2 Command Prompts panel**

   ![DB2 Command Prompts panel]

   Select command model==> Then press Enter.

   1. ALTER BUFFERPOOL
   2. ALTER GROUPBUFFERPOOL
   3. ALTER UTILITY
   4. ARCHIVE LOG
   5. CANCEL THREAD
   6. DISPLAY ARCHIVE
   7. DISPLAY BUFFERPOOL
   8. DISPLAY DATABASE
   9. DISPLAY DDF
   10. DISPLAY FUNCTION SPEC
   11. DISPLAY GROUP
   12. DISPLAY GROUPBUFFERPOOL
   13. DISPLAY LOCATION
   14. DISPLAY LOG
   15. DISPLAY PROCEDURE
   16. DISPLAY RLIMIT
   17. DISPLAY THREAD
   18. DISPLAY TRACE
   19. DISPLAY UTILITY
   20. MODIFY TRACE
   21. RECOVER BSDS
   22. RECOVER INDUBT
   23. RECOVER POSTPONED
   24. RESET GENERICLU
   25. RESET INDUBT
   26. SET ARCHIVE
   27. SET LOG
   28. SET SYSPARM
   29. START DATABASE
   30. START DDF
   31. START FUNCTION SPEC
   32. START PROCEDURE
   33. START RLIMIT
   34. START TRACE
   35. STOP DATABASE
   36. STOP DDF
   37. STOP FUNCTION SPEC
   38. STOP PROCEDURE
   39. STOP RLIMIT
   40. STOP TRACE
   41. TERM UTILITY

   * Note - These commands have no variable parameters and will be executed when they are selected from this panel.

2. In the Select command model field, type the number of the DB2 command that you want to create and press Enter.

   CATALOG MANAGER displays the command prompt panel for the command. For example, if you selected the START DATABASE command, CATALOG MANAGER displays the panel shown in Figure 129.
3 Specify the values for the DB2 command.

A In the Process type field, type N, G, or E to specify the method of processing the command, as shown in Table 51.

Table 51 Process type options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>No action</td>
</tr>
<tr>
<td>G</td>
<td>Generate the syntax, but do not execute it. The syntax that CATALOG MANAGER generates is commented out.</td>
</tr>
<tr>
<td>E</td>
<td>Generate the syntax, and immediately execute it.</td>
</tr>
</tbody>
</table>

B Specify values for the keywords for the command.

C To specify additional keywords that are not displayed on the panel, type the keyword and value in the Additional options field.

NOTE
Regardless of the command that you select, CATALOG MANAGER provides the Process type field as the first field on the panel.
D To save the command values in a profile, on the Command line, type `PROFILE SAVEAS profileName`.

CATALOG MANAGER displays a message that specifies the location of your profile. Press Enter to clear the message.

For more information about DB2 command profiles, see “DB2 command profiles” on page 318.

E Press Enter.

CATALOG MANAGER displays the DB2 Commands panel (Figure 130). The command that you generated is displayed in the Current Command line. The Command Results section displays the result of processing:

- If you selected G in step 3A, CATALOG MANAGER displays the message, “NO COMMAND EXECUTED YET.”

To execute the command, on the Current Command line, change the asterisk (*) to a dash (-) and press Enter.

- If you selected E in step 3A, CATALOG MANAGER displays the result of the executed command.

**Figure 130 DB2 Commands panel**

```
ACTPKDUB ----------------------  DB2 Commands  --------------- Row 1 to 1 of 1
Command ===>                                                  Scroll ===> PAGE
01
Model Commands    1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
Current Command  . .  *START DATABASE(PBXJ2310)
------------------------------- Command Results -------------------------------
NO COMMAND EXECUTED YET
******************************************************************************
```
Using model commands

CATALOG MANAGER displays the DB2 commands that you generate and execute on the DB2 Commands panel (Figure 130). From this panel, you can perform the following tasks:

- execute a generated command
- save the current command
- delete model commands
- change the format of the panel

To display the model commands

To display the DB2 Commands panel, from the Primary Menu, type `DB2COMMAND` on the Command line.

To execute a generated command

To execute the command, on the `Current Command` line, change the asterisk (*) to a dash (-) and press Enter.

To save the current command

To save the current command, in the `Model Commands` line, type `KEEP`.

CATALOG MANAGER copies the command from the `Current Command` field and displays it on a model commands line. The product also saves the command for the model commands in your ISPF profile.

To delete model commands

To remove the command, in the `Model Commands` line, type over the command with blanks.

To change the format of the panel

From the `Command` line on the DB2 Commands panel, you can enter the commands shown in Table 52 to change the format of the panel. CATALOG MANAGER saves these commands in your ISPF profile.
### Specifying command syntax

If you are already familiar with the syntax for a DB2 command, you can enter the command directly from the following locations:

- list line
- mixed list line
- Command line

You can also enter the command as input to batch job.

**Figure 131** shows the syntax for a DB2 command.

**Figure 131  DB2 command syntax**

```
cmd type identifier keywords [-profile] BATCH
```

**NOTE**

Some DB2 commands do not use the *type* or *identifier* variables.

Table 53 describes the variables and optional keyword for the command.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANY</td>
<td>Displays 10 lines for the model commands</td>
</tr>
<tr>
<td>FEW</td>
<td>Displays 3 lines for the model commands</td>
</tr>
<tr>
<td>NONE</td>
<td>Displays no lines for the model commands</td>
</tr>
<tr>
<td>AUTO</td>
<td>CATALOG MANAGER determines the number of model commands to display based on the amount of space needed for the command results</td>
</tr>
<tr>
<td>END</td>
<td>Returns you to the Primary Menu</td>
</tr>
</tbody>
</table>

---

Table 52  Commands used on DB2 Commands panel
### Table 53  DB2 command variables

<table>
<thead>
<tr>
<th>Variable or keyword</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cmd</strong></td>
<td>specifies the DB2 command</td>
<td>For a list of DB2 commands, see Appendix H, “Commands.”</td>
</tr>
<tr>
<td><strong>type</strong></td>
<td>specifies the object type of the <code>identifier</code></td>
<td>When the command is entered from a list line, the value of <code>type</code> is obtained from the list line. If you specify a value for <code>type</code>, CATALOG MANAGER treats the value as a keyword and generates an error. When the command is entered from the Command line or as input to a batch job, the value of <code>type</code> is required (if it applies to the command).</td>
</tr>
<tr>
<td><strong>identifier</strong></td>
<td>specifies the fully qualified name of the object</td>
<td>When the command is entered from a list line, the value of <code>identifier</code> is obtained from the list line. If you specify a value for <code>identifier</code>, CATALOG MANAGER treats the value as a keyword and generates an error. When the command is entered from the Command line or as input to a batch job, the value of <code>identifier</code> is required (if it applies to the command).</td>
</tr>
<tr>
<td><strong>keywords</strong></td>
<td>specifies additional optional values for the command</td>
<td>To display the command prompt panel for the command, use <code>?</code> as the keyword. For more information about using command prompt panels, see “Using command prompts” on page 312.</td>
</tr>
<tr>
<td><strong>-profile</strong></td>
<td>specifies the name of an existing DB2 command profile</td>
<td>For more information about using profiles, see “DB2 command profiles” on page 318.</td>
</tr>
<tr>
<td><strong>BATCH</strong></td>
<td>specifies to execute the command in batch</td>
<td>If specified, the BATCH keyword must be the last word in the syntax for the command. For more information about using the BATCH command, see “Using the BATCH command for a DB2 object list or a mixed list” on page 210.</td>
</tr>
</tbody>
</table>
You can save your customized syntax values for a DB2 command in a DB2 command profile. If you use the PROFILE command to save your syntax to the profile, CATALOG MANAGER stores the profile ID, the DB2 command keywords, and the keyword values in a utility profile. Using the profile can save time and limit errors: you no longer need to specify syntax every time you execute the command. (For more information about utility profiles, see “Utility profiles” on page 292.)

Figure 132 shows the syntax of the PROFILE command.

Figure 132  PROFILE command syntax

PROFILE [SAVE | SAVEAS profileID | SET profileID]

The options for the command are as follows:

- SAVEAS profileID, which saves the profile to a specific profile

  You can use a maximum of 64 characters, including spaces, in the profile ID. The characters `<`, `>`, `?`, and `/` are not permitted.

- SET profileID, which establishes a profile

  Using the PROFILE SET command to establish a profile will reset your current profile to the values defined in the profile.

- SAVE, which saves the DB2 command profile

  After you establish a DB2 command profile with the PROFILE SET command, you can issue the PROFILE SAVE command to save the profile.

Figure 133 shows an example of the RDACRJSTARTDB DB2 command profile in a utility profile.

Figure 133  DB2 command profile (part 1 of 2)

```xml
<?xml version="1.0"?>
<profiles>
  <profiles>
    <timestamp>12/12/04 16:37</timestamp>
    <updated_by>RDACRJ</updated_by>
    <profile>
      <process>COPY</process>
      <utility_id>COPYDEFAULT</utility_id>
      <full> </full>
  </profile>
</profiles>
```
Figure 133  DB2 command profile (part 2 of 2)

```xml
<profile>
  <process>DB2C_START_DB</process>
  <clone>N</clone>
  <access>RW</access>
  <extra> </extra>
  <profile_id>RDACRJSTARTDB</profile_id>
</profile>
</profiles>
```
Where to go from here

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</td>
<td>Chapter 11, “Managing authorizations”</td>
</tr>
<tr>
<td>Primary Menu panel or an object list panel.</td>
<td></td>
</tr>
<tr>
<td>Create and edit session profiles, which customize specific CATALOG</td>
<td>Chapter 12, “Customizing CATALOG MANAGER</td>
</tr>
<tr>
<td>MANAGER displays and operations for specific users or groups of users.</td>
<td>command access”</td>
</tr>
<tr>
<td>Browse and purge logs that CATALOG MANAGER maintains.</td>
<td>Chapter 13, “Maintaining logs”</td>
</tr>
</tbody>
</table>
Managing authorizations

This chapter contains the following topics:

Overview ................................................................. 321
Granting privileges by using the GRANT commands ............... 322
  Granting privileges on a table ........................................ 323
  Granting privileges on a hierarchy of DB2 objects ............... 327
  Importing the SQL in another subsystem .......................... 330
  Copying an SQL_Table entry to another subsystem .............. 332
Granting privileges by issuing the COPYAUTHS command ........ 333
  Copying user ID privileges ......................................... 333
  Copying object privileges ........................................... 337
Granting privileges by generating SQL ............................... 337
Revoking privileges ..................................................... 338
  Preserving access to information ................................. 339
  Generating the cascade report .................................... 339
  Revoking privileges on specific objects ......................... 342
  Verifying current authorizations ............................... 348
Where to go from here .................................................. 349

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 94.

**NOTE**
For more information, view the Quick Course Managing Authorizations. You must have a BMC Support ID to view the Quick Course.

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.

**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**

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 134). You can specify any number of tables.

   To grant privileges on every listed table, on the Command line, type GRANT ALL.
Granting privileges on a table

**Figure 134  Table List panel with GRANT command**

<table>
<thead>
<tr>
<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>GRANTZU.QZUT01_DACS03</td>
<td>QZUDAC</td>
<td>QZUS03AC</td>
<td>21</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DACS04</td>
<td>QZUDAC</td>
<td>QZUS04AC</td>
<td>21</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DACS05</td>
<td>QZUDAC</td>
<td>QZUS05AC</td>
<td>21</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DACS06</td>
<td>QZUDAC</td>
<td>QZUS06AC</td>
<td>21</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DACS07</td>
<td>QZUDAC</td>
<td>QZUS07AC</td>
<td>21</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DACS08</td>
<td>QZUDAC</td>
<td>QZUS08AC</td>
<td>21</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DACS09</td>
<td>QZUDAC</td>
<td>QZUS09AC</td>
<td>21</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DA1S01</td>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>21</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DA1S02</td>
<td>QZUDA1</td>
<td>QZUS02A1</td>
<td>21</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DA1S03</td>
<td>QZUDA1</td>
<td>QZUS03A1</td>
<td>21</td>
<td>1</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DA1S04</td>
<td>QZUDA1</td>
<td>QZUS04A1</td>
<td>20</td>
<td>0</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DA1S05</td>
<td>QZUDA1</td>
<td>QZUS05A1</td>
<td>2</td>
<td>0</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DB1S01</td>
<td>QZUDB1</td>
<td>QZUS01B1</td>
<td>11</td>
<td>0</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DB1S02</td>
<td>QZUDB1</td>
<td>QZUS02B1</td>
<td>11</td>
<td>0</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DB1S03</td>
<td>QZUDB1</td>
<td>QZUS03B1</td>
<td>11</td>
<td>0</td>
<td>T</td>
</tr>
<tr>
<td>QZU.QZUT01_DB1S04</td>
<td>QZUDB1</td>
<td>QZUS04B1</td>
<td>11</td>
<td>0</td>
<td>T</td>
</tr>
</tbody>
</table>

3 Press Enter.

The Grant Table Privileges panel is displayed (Figure 135).

**Figure 135  Grant Table Privileges panel**

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZU.QZUT01_DACS03</td>
<td>--------</td>
</tr>
</tbody>
</table>

*************************************************************************** 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 136).
Granting privileges on a table

Figure 136 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.
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 137**). 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:

4  **(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

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 327. 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 138).

**Figure 138 Import SQL from a PDS panel**

```
--- Import SQL from a PDS ---

<table>
<thead>
<tr>
<th>Command</th>
<th>Import SQL from a PDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>Member name or pattern for members to be copied</td>
</tr>
<tr>
<td>Member pattern</td>
<td>Member name or pattern for members to be copied</td>
</tr>
<tr>
<td>SQL Member owner</td>
<td>Owner of member in SQL Table</td>
</tr>
<tr>
<td>SQL Member prefix</td>
<td>Member name prefix for SQL Table</td>
</tr>
<tr>
<td>Overwrite like names</td>
<td>(Y/N Y-overwrite members with the same name)</td>
</tr>
</tbody>
</table>
```

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.
Copying an SQL_Table entry to another subsystem

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.
On the Command line, type `PASTE membername`.

---

**TIP**

To view the source SQL and membername, type `CLIPBOARD` on the Command line. Then press Enter. Press END to display the SQL_Table panel.

---

Press Enter.

The remote SQL_Table entry is pasted from the Clipboard to the CATALOG MANAGER `sqlid.BMCCAT.WORK` file.

10 (optional) Edit the SQL as needed.

11 Press END to save the SQL and exit the ISPF edit session.

---

**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 139).

**Figure 139 Copy User Authorizations panel**

![Copy User Authorizations panel]

**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 140).

**Figure 140  Confirm SQL panel for copying authorizations**

```
DEFF-R                         Confirm SQL                          1 to 5 of 5
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) . . . . . . . . .

Execute SQL. . . . . . . . . N       Y/N Execute the SQL

-------------------------------------  SQL  -----------------------------------
GRANT DBADM
    ON DATABASE
    QZUDSC30
TO RDACRJ ;

***************************************************************************** Bottom of data *****************************************************************************
```
11 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 330.
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 333.

### 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 209 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 54 lists the valid source objects for the DCL command.

Table 54  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>MQT</td>
<td>materialized query tables</td>
</tr>
<tr>
<td>NP</td>
<td>native SQL stored procedures</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>SE</td>
<td>identity columns in sequences</td>
</tr>
<tr>
<td>SG</td>
<td>storage group</td>
</tr>
<tr>
<td>SU</td>
<td>system privileges for AUTHIDs</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.
Preserving access to information

- 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.

For DB2 Version 10 and later, the REVOKE command supports the following clauses for the REVOKE DEPENDENT PRIVILEGES parameter:

- **NOT INCLUDING DEPENDENT PRIVILEGES** clause, which allows you to revoke a privilege or authority from a user, while retaining all the grants that the user has made.

  When this clause is included in the REVOKE command and you issue the CASCADE command, the CASCADE command generates a message that states that dependent privileges will not cascade.

- **INCLUDING DEPENDENT PRIVILEGES** clause, which allows you to revoke a privilege or authority from a user, while removing all the privileges or authorities that the privilege or authority granted.

**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. The report also indicates the IDs of the current installation system administrators and notes whether the ID being revoked was previously an installation system administrator.
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.

**NOTE**

The BATCH keyword is valid for the CASCADE 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 209.

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 141**) shows which privileges are affected by the REVOKE action.

**Figure 141 Cascade Report**

```sql
DEGA-R Cascade Report 1 to 12 of 28
Command ===>  Scroll ===> PAGE

Current SQLID. . . . . . . .  RDACRJ
Edit Options . . . . . . . .  N  Y/N Modify SQL processing options
Edit the Cascade Report. . .  N
Save in PDS. . . . . . . . .  N  Y/N Save SQL in PDS
PDS(member) . . . . . . . .  
Save in SQL table. . . . .  N  A/Y/R/N A/Y-Append, R-Replace
Name of saved SQL . . . .  20080918_145823

-------------------------- Report of Cascaded Grants --------------------------
Grantee  Grantor  WithGrant  Type  Name
Privs     Option
--------------------------
-- INSTALL SYSADM1: CSTCXN  SYSADM2: BMCADM
-- DEM AS GRANTOR ON 2010-09-17-10.23.39.795807
-- DEM AS GRANTEE ON 2010-08-11-16.34.35.323882
-- ASSUMING DEM WAS NEVER INSTALL SYSADM
--
DEM0809A  DEM  VW  DEM0809A.VW04A1 WILL BE DROPPED
DEM0809A  DEM  VW  DEM0809A.VW0719 WILL BE DROPPED
DEM0917A  DEM  DT  DAVL#1.UDOC5M USAGE ON DISTINCT TYPE
```
3 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 142 shows the user privileges for a database.

Figure 142 Object privileges panel
To generate a user authorizations list for that object, type UA, and then press Enter (Figure 143).

Figure 143  User List panel

The columns on an object privileges panel or a User Authorizations 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 144).
Revoking privileges on specific objects

Figure 144  Confirm SQL for Revoke Reassign panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Confirm SQL for Revoke Reassign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current SQLID</td>
<td>RDACRJ2</td>
</tr>
<tr>
<td>Reassign GRANTOR</td>
<td></td>
</tr>
<tr>
<td>Edit Options</td>
<td>N</td>
</tr>
<tr>
<td>Edit SQL</td>
<td>N</td>
</tr>
<tr>
<td>Reassign Grants</td>
<td>N</td>
</tr>
<tr>
<td>View Cascade Report</td>
<td>N</td>
</tr>
<tr>
<td>Save in SQL table</td>
<td>N</td>
</tr>
<tr>
<td>Name of saved SQL</td>
<td>20110121_115552</td>
</tr>
<tr>
<td>Save in PDS</td>
<td>N</td>
</tr>
<tr>
<td>PDS(member)</td>
<td>ACT.V10,DATABASE(TEST)</td>
</tr>
<tr>
<td>Execute SQL</td>
<td>N</td>
</tr>
<tr>
<td>REVOKE DBADM ON DATABASE ACTADMN1 FROM CATDB1 BY RDAPXB ;</td>
<td></td>
</tr>
</tbody>
</table>

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.
Revoking privileges on specific objects

**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 145).
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 146). For information about the Cascade List Report, see “Generating the cascade report” on page 339.
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.
**Verifying current authorizations**

Whether you generate the cascade report online or in batch mode, the report displays authorization information that existed in the DB2 catalog at the time that you generated the report. Because different grantors can grant a user the same authorizations, BMC recommends that you complete the following steps to ensure that the cascade report reflects authorizations that are currently granted before you revoke privileges.

1. Generate a cascade report for *User1*.

   **TIP**

   To generate a cascade report, see page 339.

2. If no privileges are affected by the REVOKE action, issue the REVOKE command for *User1*.

3. Generate a cascade report for *User2*.

4. If CATALOG MANAGER generates the report, issue the REVOKE command for *User2* and reassign the grants to an existing user.

   **TIP**

   To reassign the grants, see page 342.

5. To revoke the privileges of additional users, repeat step 1 through step 4.
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 12, “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 ................................................................. 351
Preparing to implement session profiles .................................. 352
Creating session profiles ...................................................... 352
  Commands to create and edit session profiles ............................ 353
  Creating a session profile with a customized primary menu .......... 354
  Displaying the session profiles list .................................... 356
  Displaying session profile descriptions .................................. 357
  Creating a session profile with a tailored commands table .......... 358
  Creating a session profile with an initial list filter .................... 359
Editing session profiles ..................................................... 360
  Deleting session profiles ............................................... 360
  Editing a customized Primary Menu .................................... 361
  Editing a tailored commands table .................................... 362
  Retrieving an initial list filter ........................................ 363
Assigning session profiles to users .......................................... 363
  Determining a user's capabilities ....................................... 363
  Activating and deactivating session profiles ......................... 365
Where to go from here .................................................... 366

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 55.
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 55 in a session profile.

### Table 55  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 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 table.</td>
</tr>
<tr>
<td>initial list filter</td>
<td>applies saved search variables by default when the user generates a list for 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 56 to customize the Primary Menu and tailor the commands table when creating or editing a session profile.

<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>
<tr>
<td></td>
<td>- On the Profiles List panel, enter TA in the Cmd column to edit an existing session profile.</td>
</tr>
</tbody>
</table>
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 147).

**Figure 147  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 148).

Figure 148  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 149).

**Figure 149  Profiles List panel**

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 356.

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 150).

**Figure 150  Describe Profile Entries panel**

```
DEFF-R ------------------  Describe Profile Entries  -------  Row 1 to 15 of 15
Command ===>                                                  Scroll ===> CSR

-------------------------------------------------------------------------------
FROM ACT101.SEARCH_VARS
-------------------------------------------------------------------------------
Profile . . . : LACA
Title . . . . : LACA
Main menu . . : Y
Command table :
DB filter . . :                        AL filter . . :                        LO filter . . :
TS filter . . :                        XT filter . . :                        PR filter . . :
TB filter . . :                        SY filter . . :                        ST filter . . :
VW filter . . :                        SG filter . . :                        TR filter . . :
PL filter . . :                        SU filter . . :                        FN filter . . :
PG filter . . :                        US filter . . :                        DT filter . . :
IX filter . . :                        DN filter . . :
CO filter . . :                        CK filter . . :
-------------------------------------------------------------------------------
******************************* Bottom of data ********************************
```

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).
- **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 356), and then press Enter.

  The Profile Command Tailoring List panel is displayed (Figure 151).

4 Press END.

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 356), and then press Enter.

  The Profile Command Tailoring List panel is displayed (Figure 151).
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 184.

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 186.

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.profileName 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.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 152).

   **Figure 152** Menu Profile Customizing panel to edit a session profile

   ![Menu Profile Customizing panel](image)

   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.
3 In the **Preview the profile menu** field, type **Y** to preview the customized menu. Press **Enter**.

The Preview panel for the customized Primary Menu is displayed.

A Examine the Preview Menu panel and note any desired changes.

B To display the Menu Profile Customizing panel, press **Enter** or **END**.

4 In the **Save the profile menu** field, type **Y** to save the edited menu.

5 Press **Enter**.

CATALOG MANAGER displays the Menu Profile Customizing panel with the confirmation message `PROFILE. profileName UPDATE`.

---

**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 186. 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 System User Guide and the BMC Products and Solutions for DB2 Configuration 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 57 describes how combinations of these factors determine user capabilities.

**Table 57  User capabilities determined by CRS DOPT and PR parameter**

<table>
<thead>
<tr>
<th>CRS option</th>
<th>PR parameter</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRS=Y</td>
<td><code>profileName</code></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 <code>profileName</code> and SET PROFILE OFF commands.</td>
</tr>
<tr>
<td></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 <code>profileName</code> and SET PROFILE OFF commands.</td>
</tr>
<tr>
<td>CRS=N</td>
<td><code>profileName</code></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 <code>profileName</code> and SET PROFILE OFF commands.</td>
</tr>
<tr>
<td></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 <code>profileName</code> 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 352.

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 153 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.
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 13, “Maintaining logs,” to learn about these helpful CATALOG MANAGER tools.
Maintaining logs

This chapter contains the following topics:

Overview .......................................................... 367
Accessing the logs ............................................. 368
Using the Session Log ....................................... 369
   Browsing the Session Log .............................. 369
   Purging the Session Log ............................... 372
Using the DDL Audit Log ................................. 374
Using the Drop Recovery Log ......................... 374
   Browsing the Drop Recovery Log ................. 375
   Purging the Drop Recovery Log .................. 378
Where to go from here ................................. 378

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 154).

NOTE
For more information, view the Quick Course Using Logs. You must have a BMC Support ID to view the Quick Course.

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)`.

2. On the Log Maintenance Menu panel, select **Browse Session Log**.

   The Browse Session Log panel is displayed (Figure 155).

**Figure 155  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><code>yyyy-mm-dd-hh.mm.ss.nnnnnn</code></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
Browsing the Session Log

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.

**NOTE**

If you do not indicate any specifications, CATALOG MANAGER attempts to display the entire 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 156).
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 157), which shows information from the Session Log.
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 158).

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.
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 369 and “Purging the Session Log” on page 372.

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 159).

**Figure 159  Browse Recovery Log panel**

```
DEFF-R ---------------------  Browse Recovery Log  ------------------------------
Command ====>

Type browse specifications and press Enter.

<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>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>
<tr>
<td>Product</td>
<td>. . . . =</td>
<td></td>
</tr>
</tbody>
</table>

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 160).

**Figure 160  Recovery Log List**

<table>
<thead>
<tr>
<th>Command</th>
<th>Date</th>
<th>Time</th>
<th>AuthId</th>
<th>Product</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF-R</td>
<td>2011-02-01</td>
<td>14.43</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>DBXNAUT.SBXNCOL</td>
</tr>
<tr>
<td></td>
<td>2011-01-27</td>
<td>16.03</td>
<td>RDAPXB3</td>
<td>AEX010</td>
<td>DB</td>
<td>PLBNAUT</td>
</tr>
<tr>
<td></td>
<td>2011-01-19</td>
<td>16.25</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>PLBDAL1.PLBALT02</td>
</tr>
<tr>
<td></td>
<td>2011-01-19</td>
<td>14.09</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>PLBDAL1.PLBALTER</td>
</tr>
<tr>
<td></td>
<td>2011-01-17</td>
<td>14.16</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>PXBQXPB.HASHPXBY1</td>
</tr>
<tr>
<td></td>
<td>2011-01-17</td>
<td>14.14</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TB</td>
<td>PXBTMPDB.HASH_BY_RANGE_NOT_</td>
</tr>
<tr>
<td></td>
<td>2011-01-07</td>
<td>15.52</td>
<td>RDAPXB2</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5.MXSS01G5</td>
</tr>
<tr>
<td></td>
<td>2011-01-07</td>
<td>15.46</td>
<td>RDAPXB2</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5.MXSS01G5</td>
</tr>
<tr>
<td></td>
<td>2010-12-15</td>
<td>13.12</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>QCHD14.PXBS0214</td>
</tr>
<tr>
<td></td>
<td>2010-11-05</td>
<td>15.52</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>DB</td>
<td>PXBQXPB</td>
</tr>
<tr>
<td></td>
<td>2010-09-14</td>
<td>17.22</td>
<td>RDAPXB3</td>
<td>ACT010</td>
<td>TB</td>
<td>QCH_LONG.QCH_LONG_NAME_TABL</td>
</tr>
<tr>
<td></td>
<td>2010-07-29</td>
<td>17.53</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TB</td>
<td>ACTB.CORTAPST</td>
</tr>
<tr>
<td></td>
<td>2010-04-20</td>
<td>11.20</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>ACTIBPBX.ACTSPXB1</td>
</tr>
<tr>
<td></td>
<td>2010-04-20</td>
<td>11.09</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>ACTQX13.ACTS0113</td>
</tr>
<tr>
<td></td>
<td>2010-04-20</td>
<td>11.07</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>ACTQX13.ACTS0113</td>
</tr>
<tr>
<td></td>
<td>2010-04-19</td>
<td>15.37</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>ACTQV10.ACTSV010</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 161).
In the **Cmd** column for the appropriate items, type **D** or **S** to view detailed information about entries on the list.

**9** Press Enter.

The Describe Audit Log Entry panel is displayed (Figure 162), which shows information from the Drop Recovery Log.
Press END to move to the next Describe Audit Log Entry panel (if you selected multiple items) or back to the Recovery Log Detail panel.

Purging the Drop Recovery Log

For information about purging the Drop Recovery Log, see “Purging the Session Log” on page 372.

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 ............................................. 379
Modifying the ISPF skeletons and variables ............................ 380
Sizing JCL Generation data sets ........................................... 381

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 generates JCL by using a method that improves upon
standard ISPF file-tailoring services. The method used is based on ISPF skeletons. 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 58. 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 131.
Table 58  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 61 on page 403 and Table 62 on page 409 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, BMC uses an SLIB compiler. Consequently, *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. You determine the source to use on the JCL Generation Static Data Set Options panel.

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 59 shows the statistics that JCL Generation uses for space estimation and the source of the statistics.
Table 59  Data set sizing values and sources

<table>
<thead>
<tr>
<th>Value</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BMCSTATS</td>
</tr>
<tr>
<td>number of active pages</td>
<td>X</td>
</tr>
<tr>
<td>number of modified pages</td>
<td>X</td>
</tr>
<tr>
<td>page size</td>
<td>NA</td>
</tr>
<tr>
<td>maximum row length</td>
<td>NA</td>
</tr>
<tr>
<td>average row length</td>
<td>X</td>
</tr>
<tr>
<td>number of rows</td>
<td>X</td>
</tr>
<tr>
<td>number of non-clustering indexes</td>
<td>NA</td>
</tr>
<tr>
<td>longest key</td>
<td>X</td>
</tr>
<tr>
<td>number of foreign keys</td>
<td>NA</td>
</tr>
<tr>
<td>number of indexes</td>
<td>X</td>
</tr>
<tr>
<td>longest foreign key</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 104.
Using the Skeleton Library compiler

This appendix includes the following topics:

Overview ................................................................. 383
Compiling SLIBs ...................................................... 384
Changing SLIBs ........................................................ 385
  Testing changes using ISPF file tailoring ...................... 386
  Compiling changed SLIBs ....................................... 387
Processing SLIBs ....................................................... 387
  Generating the SLIB report ..................................... 387

Overview

BMC has improved the performance of JCL construction by using the BMC 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 163 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 SLIB changes by using standard ISPF file tailoring before you compile the SLIBs. For details, see “Testing changes using ISPF file tailoring” on page 386.

# 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.

**WARNING**
BMC uses SMP/E to package and deliver the SLIB members. If you must make a change to an SLIB, consider copying the SLIB member or members to be changed into a separate library. You can then make your changes to the SLIB members in that library without the risk of applying SMP/E maintenance that overlays your changes. Note that PTFs or GA releases that BMC distributes might change the SLIB source. You will need to determine whether the SLIB source was changed. If the SLIB source was changed, you will need to copy the new version of the SLIB source to your separate library, and reapply any changes that you made.
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:

  `)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, $USESTFT, is allocated as DUMMY to either the TSO session or the batch job that is being executed.
  
  Add the ddname $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:

  `//$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:

  `TSO ALLOC FI($USESTFT) DA('NULLFILE') SHR`

  The option remains in effect until you log off or you issue the following command:

  `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 JCM NO-COMPILE option if you used it. Before you test the compiled SLIB, turn off any other options that you used, such as $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 387.

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.

   LRECL=80,RECFM=FB, BLKSIZE=6160,DSORG=PS

2. Resubmit your job.

   An example follows:

   ```
   //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:

   ```
   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 164 shows a sample runtime report.

   **Figure 164  Sample runtime report (Part 1 of 2)**

<table>
<thead>
<tr>
<th>Skelname</th>
<th>Usage</th>
<th>Compile Type</th>
<th>Date</th>
<th>Time</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJX$ACMX</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.07</td>
<td>1</td>
</tr>
<tr>
<td>AJXJOB0</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.42</td>
<td>1</td>
</tr>
<tr>
<td>AJX#USRV</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.39</td>
<td>1</td>
</tr>
<tr>
<td>AJXJOB5</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.37</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTEP1</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>09.50</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTEP7</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.00</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTEPU</td>
<td></td>
<td>Compiled</td>
<td>01/29/2013</td>
<td>13.50</td>
<td>1</td>
</tr>
<tr>
<td>AJXSYSX$</td>
<td></td>
<td>Compiled</td>
<td>01/29/2013</td>
<td>17.09</td>
<td>1</td>
</tr>
<tr>
<td>AJXSYSMD</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>11.05</td>
<td>1</td>
</tr>
<tr>
<td>AJXSTWK0</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>11.01</td>
<td>1</td>
</tr>
<tr>
<td>AJXSYSST</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.47</td>
<td>1</td>
</tr>
<tr>
<td>AJXISPFM</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>09.51</td>
<td>1</td>
</tr>
<tr>
<td>AJXCLIBU</td>
<td></td>
<td>Compiled</td>
<td>01/29/2013</td>
<td>17.09</td>
<td>1</td>
</tr>
<tr>
<td>AJXMLIBU</td>
<td></td>
<td>Compiled</td>
<td>01/29/2013</td>
<td>17.09</td>
<td>2</td>
</tr>
<tr>
<td>AJXISPSFS</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.09</td>
<td>1</td>
</tr>
<tr>
<td>AJXSLIBU</td>
<td></td>
<td>Compiled</td>
<td>01/29/2013</td>
<td>17.09</td>
<td>1</td>
</tr>
<tr>
<td>AJXTLIBU</td>
<td></td>
<td>Compiled</td>
<td>01/29/2013</td>
<td>17.09</td>
<td>1</td>
</tr>
<tr>
<td>AJXPLIBU</td>
<td></td>
<td>Compiled</td>
<td>01/29/2013</td>
<td>17.09</td>
<td>1</td>
</tr>
<tr>
<td>AJX#PRNT</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>11.13</td>
<td>1</td>
</tr>
<tr>
<td>AJXNOSTS</td>
<td></td>
<td>Compiled</td>
<td>01/19/2013</td>
<td>10.56</td>
<td>1</td>
</tr>
</tbody>
</table>
The report summary at the end of Figure 164 provides the information shown in Table 60.

**Table 60  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 the Common Explain component

This appendix contains the following topics:

Overview ................................................................. 391
Before you begin ....................................................... 392
Commands to access SQL ........................................... 392
  Explaining a DBRM, package, or plan ......................... 394
  Explaining a statement from a DBRM or package .......... 395
  Explaining a statement from the SQL_Table ................ 398
  Editing the SQL statement by using the SQLX Edit Macro. 400

Overview

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 the Common Explain component of the BMC Software SQL Explorer for DB2 and the BMC Software APPTUNE for DB2 products. For detailed information about the functions of Common Explain and the reports that it produces, see the SQL Explorer for DB2 User Guide.
Before you attempt to use the Common Explain component with CATALOG MANAGER, verify that the ACTPSS CLIST has been installed and customized. For more information, see the Installation System User Guide and the BMC Products and Solutions for DB2 Configuration 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 165).

Figure 165 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 166).

**Figure 166 Using the DESCRIBE and GET commands**
Explaining a DBRM, package, or plan

In this task, you use the BMCEXPLORE command to access the Common Explain component 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 168).
Explaining a statement from a DBRM or package

In this task, you use the DESCRIBE command to access the Common Explain component 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.

Figure 168 BMCEXPLORE command issued against plan

3. Press Enter to begin the Explain processing. For more information, see the SQL Explorer for DB2 User Guide.
Explaining a statement from a DBRM or package

**3** Press Enter.

The DESCRIBE panel for the source object is displayed (Figure 169).

**Figure 169** DESCRIBE panel

---

**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 170).
Explaining a statement from a DBRM or package

Figure 170  Use of GET subcommand

```
DEFF-R Package: DSN_DEFAULT_COLLID_QZUTSTPL.QZUTSTPL Line 22 of 42 Col 1 80
Command === GET 24

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

DEFF-R Package: DSN_DEFAULT_COLLID_QZUTSTPL.QZUTSTPL Line 22 of 42 Col 1 80
Command === GET 24

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

Stmntno 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
```

6 Press Enter.

The Confirm SQL panel is displayed (Figure 171).

Figure 171  Confirm SQL panel

```
DEFF-R Confirm SQL 1 to 3 of 3
Command ===>

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

**--------------------------- END OF DATA ---------------------------**

SELECT COLUMN_1
FROM QZU.QZUT01_D15S01
WHERE COLUMN_1 = 1234

************************************************************************** Bottom of data **********************************************************************************
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 400.

- To begin 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 the Common Explain component 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 217.

2 In the Command (Cmd) column beside the source statement, type ANALYZE (Figure 172).
3 Press Enter.

The Confirm SQL panel is displayed (Figure 173).

Figure 172 SQL Table List panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMD will show commands for this list. Type command and press Enter 01</td>
<td>CSR</td>
</tr>
<tr>
<td>Subcommands are: ANALYZE, CUT, DELETE, EDIT, EXECUTE, PASTE, RENAME, TB BROWSE, T BEDIT, 2WL. ANALYZE may be used with SELECT, INSERT, DELETE and UPDATE SQL.</td>
<td></td>
</tr>
<tr>
<td>SQL NAME LIKE: <em>.</em></td>
<td></td>
</tr>
<tr>
<td>Cmd</td>
<td>Owner</td>
</tr>
<tr>
<td>----</td>
<td>------</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF04S1</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF04U1</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF05D1</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF05I1</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF05I2</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF05S1</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF05S2</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF05U1</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF06D1</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF06I2</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF06U1</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF07SC</td>
</tr>
<tr>
<td>ANALYZEVJSJXL1</td>
<td>DEFF08SC</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF09SC</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF10SC</td>
</tr>
<tr>
<td>MVSJXL1</td>
<td>DEFF11SC</td>
</tr>
</tbody>
</table>

Figure 173 Confirm SQL panel

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current SQLID. . . . . . . RDACRJ</td>
<td>PAGE</td>
</tr>
<tr>
<td>Edit Options . . . . . . N</td>
<td>Y/N Modify SQL processing options</td>
</tr>
<tr>
<td>Edit SQL . . . . . . N</td>
<td>Y/N Edit SQL before executing</td>
</tr>
<tr>
<td>Save in SQL table. . . . N</td>
<td>A/Y/R/N A/Y-Append, R-Replace</td>
</tr>
<tr>
<td>Name of saved SQL . . . 20110121_115552</td>
<td></td>
</tr>
<tr>
<td>Save in PDS. . . . . N</td>
<td>Y/N Save SQL in PDS</td>
</tr>
<tr>
<td>PDS(member) . . . . . . . .</td>
<td></td>
</tr>
<tr>
<td>Analysis . . . . . . N</td>
<td>Y/N Call SQL Explorer for EXPLAIN</td>
</tr>
<tr>
<td>Edit/Browse data . . . N</td>
<td>E/B/N Call the Table Editor</td>
</tr>
<tr>
<td>Execute SQL . . . . N</td>
<td>Y/N Execute the SQL</td>
</tr>
<tr>
<td>-----------------------</td>
<td>SQL</td>
</tr>
<tr>
<td>SELECT * FROM OCHDBIL2.OCRSP03_TT31B ;</td>
<td></td>
</tr>
<tr>
<td>INSERT INTO OCHDBIL2.OCRSP03_TT31B ( SSSNO, EMPNO, EMPNAME, EMPDEPT,</td>
<td></td>
</tr>
</tbody>
</table>
4 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 400.

- To begin 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 398.

2 On the Confirm SQL panel, in the Edit SQL field, type Y.

3 Press Enter.

An ISPF edit panel is displayed (Figure 174).

**Figure 174  ISPF Edit panel**

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 Explain processing. For more information, see the SQL Explorer for DB2 User Guide.
JCL Generation keywords and variables

This appendix contains the following topics:

- AEXIN keywords .......................................................... 403
- Symbolic variables .......................................................... 409

AEXIN keywords

Table 61 lists the keywords in the AEXIN input stream.

Table 61  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>
<tr>
<td>BINDFAIL</td>
<td>causes worklist execution to stop with a return code of 8 if a bind fails</td>
</tr>
<tr>
<td></td>
<td>The halt will be noted in the sync tables, and an Execution restart will continue with the command that caused the failure.</td>
</tr>
<tr>
<td></td>
<td>Without this parameter, worklist execution continues if a bind fails.</td>
</tr>
</tbody>
</table>
AEXIN keywords

Table 61  AEXIN keywords  (Part 2 of 7)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. |
| CATUTIL       | specifies the worklist job                                                                                                                                                                                  |
| 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. |
### AEXIN Keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPYOPT</td>
<td>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.</td>
</tr>
<tr>
<td>DASDDOPT name</td>
<td>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.</td>
</tr>
<tr>
<td>DB2STMSGS</td>
<td>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.</td>
</tr>
<tr>
<td>DEBUGUNLD</td>
<td>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.</td>
</tr>
<tr>
<td>DYNWORKUNIT</td>
<td>For the Database Administration solution, defines the unit (such as SYSDA) that Execution uses to dynamically allocate temporary work data sets.</td>
</tr>
<tr>
<td>ENV</td>
<td>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.</td>
</tr>
<tr>
<td>FLOW</td>
<td>Causes Execution to produce flow trace messages in AEXPRINT when entering and exiting modules.</td>
</tr>
<tr>
<td>HASHFAIL</td>
<td>Causes Execution to terminate the job if a hash failure, such as a changed or added statement, occurs in a worklist.</td>
</tr>
<tr>
<td>HASHWARNRC returnCode</td>
<td>Defines the return code (returnCode) that Execution sends back when it finds only hash warnings. Do not use 8 for this value.</td>
</tr>
<tr>
<td>ITERATIONMODE</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Table 61  AEXIN keywords (Part 4 of 7)

<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</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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>
<tr>
<td>NOFAILNOIMAGECPY</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>■ No image copies of the object exist.</td>
</tr>
<tr>
<td></td>
<td>■ The CATRECOVER keyword is specified in the AEXIN input stream.</td>
</tr>
<tr>
<td></td>
<td>The product automatically generates the NOFAILNOIMAGECPY keyword in the AEXIN input stream when one of the following conditions exists:</td>
</tr>
<tr>
<td></td>
<td>■ The DROPR_NOIC POF keyword is set to Y.</td>
</tr>
<tr>
<td></td>
<td>■ The NOFAILNOIMAGECPY option is set to Y on the JCL Generation Debugging, Display and Execution panel.</td>
</tr>
<tr>
<td></td>
<td>This keyword is useful only if you have installed CATALOG MANAGER.</td>
</tr>
<tr>
<td></td>
<td>This keyword overrides the DROPR_NOIC keyword in the AJXPOFIN input stream.</td>
</tr>
<tr>
<td>NOLOADCOMP</td>
<td>instructs Execution not to compress extra spaces out of LOAD statements</td>
</tr>
<tr>
<td>NOSQLCOMP</td>
<td>instructs Execution not to compress extra spaces out of SQL statements</td>
</tr>
<tr>
<td>NOSTARTOVER</td>
<td>instructs Execution not to start the worklist again from the beginning of the worklist</td>
</tr>
<tr>
<td></td>
<td>See also the STARTOVER keyword in this table.</td>
</tr>
<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</td>
</tr>
<tr>
<td></td>
<td>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>
</tbody>
</table>
### Table 61  AEXIN keywords (Part 5 of 7)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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.</td>
</tr>
<tr>
<td></td>
<td>When running standard JCL, the condition code is added to the step subsequent to the REBIND step.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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>
<tr>
<td>SSID $ssid$</td>
<td>identifies the DB2 subsystem ID or the DB2 data sharing group attachment name.</td>
</tr>
<tr>
<td></td>
<td>The SSID keyword must match the -SSID command in the worklist. This parameter is required.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>The AJXIN and ALUIN input streams also use this keyword.</td>
</tr>
<tr>
<td>STARTOVER</td>
<td>instructs Execution to start the worklist again from the beginning of the worklist.</td>
</tr>
<tr>
<td></td>
<td>You cannot specify the STARTOVER keyword with the RESTART keyword. See also the NOSTARTOVER keyword in this table.</td>
</tr>
<tr>
<td>STATS</td>
<td>prints the execution statistics.</td>
</tr>
</tbody>
</table>
**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).

**STOPWTSECS n**  
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.

**SYNCDELETE**  
instructs Execution to remove all sync entries when an Execution job completes with no errors

**SYNLIST**  
prints a synonym list

**TASKID a.b**  
specifies the task ID to use

**UNLOADDOPT**  
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.
Symbolic variables

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 62 lists all of the symbolic variables that are used by the JCL Generation and Execution components. Some of these variables are not applicable to each of the BMC Administrative products. The table indicates which products resolve the variables in job cards and data set names. 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 System User Guide and the BMC Products and Solutions for DB2 Configuration Guide.

Table 62  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>
</tbody>
</table>

indicates whether JCL Generation verifies that the prefix of a data set name contains 44 characters.

| ALID\(^a\(^b\)\) product ID | 8    | ALTER CATMGR DASDMGR | AJXJAID | UTIL | UT               |
| ATTACH DB2 subsystem ID       | 4    | none                | AJXJSSID | ATTACH | JOBNAME          |
Table 62  Symbolic variables (Part 2 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>BMCCP two- to four-digit partition number in which insignificant digits are suppressed. If the partition number is between 0 and 99, the variable will resolve to 2 digits; between 100 and 999, 3 digits; and between 1000 and 4096, 4 digits.</td>
<td>2 to 4</td>
<td>none</td>
<td>AJXBMCCP</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>CR creator</td>
<td>8</td>
<td>none</td>
<td>AJXCR</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>DA DAY day part of YYMMDD format</td>
<td>2</td>
<td>DD</td>
<td>AJXYMD</td>
<td>DAY</td>
<td>DA DAY</td>
</tr>
<tr>
<td>DATEab system date (same format as JYMD and YMD variables)</td>
<td>6</td>
<td>YYMMDD</td>
<td>AJXYMD</td>
<td>DATE</td>
<td>DATE</td>
</tr>
<tr>
<td>DATEJ 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>DB DBNAME database name</td>
<td>8</td>
<td>name of current database</td>
<td>AJXDB</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>DB2V2 version of DB2</td>
<td>2</td>
<td>version of DB2</td>
<td>AJXDB2V2</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>DB2V3 version of DB2</td>
<td>3</td>
<td>version of DB2</td>
<td>AJXDB2V3</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>DDD Julian day</td>
<td>3</td>
<td>DDD</td>
<td>AJXYYDDD</td>
<td>JDAY</td>
<td>JDAY</td>
</tr>
<tr>
<td>DDNAME DDname</td>
<td>8</td>
<td>none</td>
<td>AJXJDDN</td>
<td>none</td>
<td>none</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>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 Software 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>GDGC</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>
<tr>
<td>HO</td>
<td>2</td>
<td>HH</td>
<td>AJXHMS</td>
<td>HOUR</td>
<td>HO</td>
</tr>
<tr>
<td>HOUR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hour part of HHMMSS format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IC</td>
<td>1</td>
<td>L = local R = remote</td>
<td>AJXIC</td>
<td>ICTYPE</td>
<td>IC</td>
</tr>
<tr>
<td>ICTYPE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ICTYPE</td>
</tr>
<tr>
<td>image copy type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 62  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>IS</td>
<td>8</td>
<td>none</td>
<td>AJXIXSPC</td>
<td>TS</td>
<td>TS</td>
</tr>
<tr>
<td>index space name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IX</td>
<td>16</td>
<td>name of current index</td>
<td>AJXIX</td>
<td>TS</td>
<td>IS</td>
</tr>
<tr>
<td>IXNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IXCR</td>
<td>8</td>
<td>creator of current index</td>
<td>AJXCR</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>index creator name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IXNODEd</td>
<td>22</td>
<td>none</td>
<td>AJXIXNOD</td>
<td>DB.TS</td>
<td>DB..IS</td>
</tr>
<tr>
<td>index node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IXSPC</td>
<td>8</td>
<td>none</td>
<td>AJXIXSPC</td>
<td>TS</td>
<td>IS</td>
</tr>
<tr>
<td>index space name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JD</td>
<td>3</td>
<td>YYDDD</td>
<td>AJXYYDDD</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Julian date shown with two-digit year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JDATE</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>JDAY</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>JDDN</td>
<td>8</td>
<td>none</td>
<td>AJXDDN</td>
<td>SEQ</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>ddname for skeleton</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JHMS</td>
<td>6</td>
<td>HHMMSS</td>
<td>AJXHMS</td>
<td>TIME</td>
<td>TIME</td>
</tr>
<tr>
<td>time of work ID creation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JJULD</td>
<td>5</td>
<td>YYDDD</td>
<td>AJXJJULD</td>
<td>JDAY</td>
<td>JDAY</td>
</tr>
<tr>
<td>Julian date of work ID creation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOBCHARab</td>
<td>1</td>
<td>A = Analysis</td>
<td>AJXFJCHR</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>a single character that indicates the component for which JCL is being generated</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>
</tbody>
</table>
### Table 62 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>JOBNAME</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>JOBTYPEab</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>JPCODa</td>
<td>3</td>
<td>ACM = CHANGE MANAGER ACT = CATALOG MANAGER ALU = ALTER ASU = DASD MANAGER PLUS</td>
<td>AJXJPCOD</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>JS1</td>
<td>1</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>JS2</td>
<td>2</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>JS4</td>
<td>4</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</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>JSSIDe</td>
<td>4</td>
<td>none</td>
<td>AJXSSID</td>
<td>SSID</td>
<td>SSID</td>
</tr>
<tr>
<td>JOBTYPab</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>JPCODa</td>
<td>3</td>
<td>ACM = CHANGE MANAGER ACT = CATALOG MANAGER ALU = ALTER ASU = DASD MANAGER PLUS</td>
<td>AJXJPCOD</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>JS1</td>
<td>1</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>JS2</td>
<td>2</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>JS4</td>
<td>4</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
</tbody>
</table>
## Symbolic variables

### Table 62  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>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>JUL4Ya</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>JULIANab</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 LIST</td>
</tr>
<tr>
<td>LIST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></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>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>
### Table 62  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>
</table>
| MEMBER<sub>ab</sub> MEMBR         | 8    | 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  
<sup>Note</sup>: If the work ID template is used in CM/PILOT, then the task ID is used. | AJXMEMBER AJXMEMBR | JOBNAME | JOBNAME |
| PDS member name                   |      |                                                                      |                       |                            |                                |
| MI MINUTE                         | 2    | MM                                                                  | AJXHMS                | MINUTE                     | MI MINUTE                      |
| minute part of HHMMSS format      |      |                                                                      |                       |                            |                                |
| MMDD                              | 4    | MMDD                                                                |                       | MONTH.DAY                  | MONTH.DAY                      |
| JCL date                          |      |                                                                      |                       |                            |                                |
| MO MONTH                          | 2    | MM                                                                  | AJXYMD                | MONTH                      | MO MONTH                       |
| month part of YYMMDD format       |      |                                                                      |                       |                            |                                |
| MSSID DB2 subsystem ID            | 4    | determined at runtime from the -JCLP command in the worklist        | AJXMSSID              | SSID                       | SSID                           |
| OBJT OBJTYP                       | 2    | two-character code that indicates the object type: TS, TB, IX        | AJXOBT                | UTIL TYPE                  | IC                             |
| object type                       |      |                                                                      |                       |                            |                                |
| OBNAM                             | 27   | none                                                                 | AJXOBNAM              | DB.TS                      | DB.SN                          |
| object name                       |      |                                                                      |                       |                            |                                |
| OBNOD database and either table space name or index space name | 17   | #### (for objects that do not exist)                                | AJXOBNOD              | DB.TS                      | DB.SN                          |
### Symbolic variables

#### Table 62  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>PA</td>
<td>3</td>
<td>none</td>
<td>AJXPARTC</td>
<td>PART</td>
<td>PA</td>
</tr>
<tr>
<td>partition number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PART</td>
<td>4</td>
<td>none</td>
<td>AJXPARTC</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>four-digit partition number in which insignificant digits are suppressed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PART4</td>
<td>4</td>
<td>none</td>
<td>AJX4PART</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>four-digit partition number in which leading zeros are not suppressed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PART5</td>
<td>5</td>
<td>none</td>
<td>AJX5PART</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>five-digit partition number in which leading zeros are not suppressed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PB PRIBAC</td>
<td>1</td>
<td>P = primary B = backup</td>
<td>AJXPB</td>
<td>TYPE</td>
<td>PB PRIBAC</td>
</tr>
<tr>
<td>type of copy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PGMR&lt;sup&gt;ab&lt;/sup&gt;</td>
<td>20</td>
<td>‘&amp;&amp;JOBTYP - &amp;&amp;WKID’</td>
<td>AJXPGMR</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>programmer name field</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(ALTER and CHANGE MANAGER)</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>PREFIX&lt;sup&gt;ab&lt;/sup&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSO prefix (same as ZPREFIX)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RHLQ</td>
<td>60</td>
<td>high-level qualifier (HLQ) for ISPF data sets</td>
<td>AJXRHLQ</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>RSEQ#</td>
<td>6</td>
<td>none</td>
<td>AJXRSEQ#</td>
<td>SEQ</td>
<td>SEQ</td>
</tr>
<tr>
<td>restart sequence number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTYPE</td>
<td>2</td>
<td>none</td>
<td>AJXOBJT</td>
<td>TYPE</td>
<td>IC</td>
</tr>
<tr>
<td>object type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 62  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>RUNTYP&lt;sup&gt;ab&lt;/sup&gt;</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>type of run</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td>2</td>
<td>SS</td>
<td>AJXHMS</td>
<td>SEC</td>
<td>SC</td>
</tr>
<tr>
<td>SEC</td>
<td></td>
<td></td>
<td></td>
<td>SECOND</td>
<td>SC</td>
</tr>
<tr>
<td>SECOND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SECOND</td>
</tr>
<tr>
<td>second part of HHMMSS format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQ</td>
<td>4</td>
<td>none</td>
<td>AJXDDSQC</td>
<td>SEQ</td>
<td>SEQ</td>
</tr>
<tr>
<td>DD sequence number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQ#</td>
<td>6</td>
<td>none</td>
<td>AJXSEQ#</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>SN</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>SPNAME</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>SQ</td>
<td>4</td>
<td>none</td>
<td>AJXDDSQC</td>
<td>SEQ</td>
<td>SQ</td>
</tr>
<tr>
<td>DD sequence number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS</td>
<td>4</td>
<td>none</td>
<td>AJXSSID</td>
<td>SSID</td>
<td>SS</td>
</tr>
<tr>
<td>DB2 subsystem ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSID&lt;sup&gt;ab&lt;/sup&gt;</td>
<td>4</td>
<td>determined at runtime</td>
<td>AJXSSID</td>
<td>SSID</td>
<td>SSID</td>
</tr>
<tr>
<td>DB2 subsystem ID (same as JSSID)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>8</td>
<td>none</td>
<td>AJXSTEPN</td>
<td>STEPNAME</td>
<td>STEPNAME</td>
</tr>
<tr>
<td>STEPN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STEPNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 62  Symbolic variables (Part 10 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>SYSUID</td>
<td></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>TBNAME</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 TIMEab</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 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>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>

[^]1 Symbolic variables and their descriptions.
### Table 62  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>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 IC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UCMD command ID</td>
<td>4</td>
<td>none</td>
<td>AJXUCMD</td>
<td>UTIL UT</td>
<td></td>
</tr>
<tr>
<td>UDOPT installation options file Lname</td>
<td>8</td>
<td>none</td>
<td>AJXUDOPT STEPNAME STOPNAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UID TSO user ID</td>
<td>7</td>
<td>none</td>
<td>ZUSER none</td>
<td><strong>PREFIX</strong></td>
<td></td>
</tr>
<tr>
<td>ULLQ low-level qualifier for user-defined data sets</td>
<td>4</td>
<td>none</td>
<td>AJXULLQ none none</td>
<td></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 PARTIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER1 user-defined</td>
<td>none</td>
<td>none</td>
<td>none UTIL UT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER2 user-defined</td>
<td>none</td>
<td>none</td>
<td>none UTIL UT</td>
<td></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>

For example, if a partition number is 10, AJXUPART will contain 010, while AJXPARTC will contain 10.
### Table 62  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>
<tbody>
<tr>
<td>UT UTID UTIL UTILID</td>
<td>16</td>
<td>none</td>
<td>AJXUTID</td>
<td>UTIL UTILID UTIL UTILID</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 &quot;##&quot; (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 VCAT name</td>
<td>8</td>
<td>none</td>
<td>AJXVCAT</td>
<td>DB DB</td>
<td></td>
</tr>
<tr>
<td>WKID&lt;sup&gt;b&lt;/sup&gt;</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 Software DASD MANAGER PLUS product</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WKOWN&lt;sup&gt;a&lt;/sup&gt; WKOWNER&lt;sup&gt;a&lt;/sup&gt;</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&lt;sup&gt;reg&lt;/sup&gt;</td>
<td>18</td>
<td>the name of the current work ID in use, 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</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>work ID name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Table 62  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>WORKID&lt;sup&gt;abant&lt;/sup&gt; first eight characters of the work ID name</td>
<td>8</td>
<td>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></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YE YEAR four-digit year from Julian date</td>
<td>4</td>
<td>YYYY</td>
<td>AJX4YDDD</td>
<td>YEAR YY</td>
<td>YE</td>
</tr>
<tr>
<td>YY two-digit year from Julian date</td>
<td>2</td>
<td>YY</td>
<td>AJXYYDDD</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>YMD date of JCL creation (same as DATE and JYMD)</td>
<td>6</td>
<td>YYMMDD</td>
<td>AJXYMD</td>
<td>DATE</td>
<td>DATE</td>
</tr>
<tr>
<td>YYDDD Julian date of JCL creation (same as JULIAN)</td>
<td>5</td>
<td>YYDDD</td>
<td>AJXYYDDD</td>
<td>JDATE</td>
<td>JDATE</td>
</tr>
<tr>
<td>YYYYDDD 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>ZACCTNUM user’s account number for jobs that are generated by the product</td>
<td>40</td>
<td>If the replacement value is not known or does not fit in the space provided, question marks (?) are substituted.</td>
<td>ZACCTNUM</td>
<td>USERID</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>ZPREFIX&lt;sup&gt;abant&lt;/sup&gt; TSO prefix</td>
<td>8</td>
<td>none</td>
<td>ZPREFIX</td>
<td>USERID</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>ZSYSID 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&lt;sup&gt;abant&lt;/sup&gt; user ID</td>
<td>8</td>
<td>none</td>
<td>ZUSER</td>
<td>USERID</td>
<td><strong>PREFIX</strong></td>
</tr>
</tbody>
</table>

---

<sup>a</sup> ALTER and CHANGE MANAGER resolve this variable for job cards and data set names.

<sup>b</sup> DASD MANAGER PLUS resolves this variable for job cards and data set names.
c 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.

d This value truncates after eight characters when used by JCL Generation.

e This value is determined at runtime (same as SSID).

f This value truncates after eight characters when used by JCL Generation.

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.

h Leave this variable blank for NOPREFIX (same as PREFIX).
CATALOG MANAGER installation options

This appendix presents the following topics:

Overview .......................................................... 423
Installation options ............................................. 423

Overview

The installation options module that the installation process creates for CATALOG MANAGER resides in $xnmDOPT and also in HLQ.UDBCNTL (where HLQ is the high-level qualifier) with the same member name as the installation options module. This chapter provides an example of the module and descriptions of each option. These installation options are also known as default options, or DOPTs.

Installation options

Figure 175 provides an example of the installation options module for CATALOG MANAGER.

Figure 175  CATALOG MANAGER installation options module (part 1 of 5)

**********************************************************************
* MODULE NAME : ACTDOPD1
* FUNCTION    : CATALOG MANAGER DEFAULT PROFILE MODULE
* COPYRIGHT   : COPYRIGHT BMC SOFTWARE INC., 2013
* LEVEL       : RELEASE 11.1, June 2013
* FUNCTIONS   : DEFINE THE DEFAULT PROFILE VARIABLES
*
### Installation options

#### Figure 175  CATALOG MANAGER installation options module  (part 2 of 5)

```
***********************************************************************
***********************************************************************
*  SECTIONS:                                                      *
*  ACTDOPTS CSECT                                                  *
*  ***********************************************************************
ACTDOPTS CSECT
$ACTDOPT
   DPT=('.',R),                                                 
  ESC='''',                                                   
  CUP=(Y,R),                                                  
  TRS=(N,R),                                                  
  CRS=(N,R),                                                  
  DRO=O,                                                      
  AUDIT=Y,                                                    
  ALLC=N,                                                     
  DBCS=(N,R),                                                 
  PLP=55,                                                     
  MAX=300,                                                    
  MPLAN=ACT111DM,                                             
  LPLAN=ACT111DL,                                             
  UPLAN=ACT111DU,                                             
  KPLAN=ACT111DK,                                             
  HPLAN=ACT111DH,                                             
  EPLAN=ACT111DE,                                             
  BPLAN=ACT111DB,                                             
  SPLAN=ACT111DS,                                             
  RCCOL=ACT111_D_MAIN,                                        
  ICCOL=,                                                     
  ICSYC=,                                                     
  CATOP=Y,                                                     
  PDSN=(&&ZUSER..BMCCAT.PRINT,R),                            
  WDSN=(&&ZUSER..BMCCAT.WORK,R),                             
  ADSN=(&&ZUSER..BMCCAT.ARCHIVE,R),                          
  BDSN=('''BMCADMN.V111.D10.DDBBRM''',R),                     
  LDSN=(&&ZUSER..BMCCAT.SQL,R),                              
  JDSN=(&&ZUSER..BMCCAT.JCL(),R),                            
  UODSN='''&ZUSER..BMCCAT.USEROPT''',                         
  TDSN=( ,R),                                                 
  POFDS=('BMCADMN.D10.UDBCNTL(AJXB1POF)',R),                 
  HDT=O,                                                      
  HDTB=O,                                                     
  HDAL=N,                                                     
  HDIX=Y,                                                     
  HDSY=N,                                                     
  HDVW=Y,                                                     
  HDPL=Y,                                                     
  HDTR=Y,                                                     
  TNCC=<>,                                                    
  TNLMR=M,                                                    
```
### Figure 175  CATALOG MANAGER installation options module (part 3 of 5)

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOPTS</td>
<td>(ACMDOPD1,R)</td>
</tr>
<tr>
<td>BOPTS</td>
<td>(ASUDOPD1,R)</td>
</tr>
<tr>
<td>GPLAN</td>
<td>ACT911DG</td>
</tr>
<tr>
<td>XODSN</td>
<td>(<em>BMCADM.V10.STDCUST.DBXML</em>,R)</td>
</tr>
<tr>
<td>$ACTSOLD</td>
<td>AUDPOL=DYNAMIC, X</td>
</tr>
<tr>
<td>AUTOALE</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>AUTORUN</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>AUTOWIN</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>AUXRELS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>CHECKDE</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>CHECKS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>CHECKS2</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>COLAUTH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>COLDISH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>COLDISS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>COLDIST</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>COLSTAT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>COLUMNH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>COLUMNN</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>CONSTDDE</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>CONTRLD</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>CONTEXT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>COPY</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>CXATTR</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>CXAUTH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>DATABAS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>DATATYP</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>DBAUTH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>DBRM</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>DEPEND</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>ENVIRON</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>FIELDS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>FOREIGN</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>INDEXES</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>INDEXH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>INDEXPA</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>INDEXPH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>INDEXSH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>INDEXST</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>IPLIST</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>IPNAMES</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>JARCONT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>JAROBJT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>JAVAPTH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>JAVOPTS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>KCOLUSE</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>KEYS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>KEYTDST</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>KEYTGD</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>KEYTGDH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>KEYTGHS</td>
<td>DYNAMIC, X</td>
</tr>
</tbody>
</table>
### Installation options

**Figure 175  CATALOG MANAGER installation options module (part 4 of 5)**

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEYTGST</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>KEYGT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>LOBSTAH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>LOBSTAT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>LOCATIO</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>LULIST</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>LUMODES</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>LUNAMES</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>MODESEL</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>OBDS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>OBJROL</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PACKAGE</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PACKAUT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PACKCPY</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PACKDEP</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PACKLIS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PACKSTM</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PARMS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PENDDDL</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PKSYSTE</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PLAN</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PLANAUT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PLANDEP</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PLSYSTE</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>PROCEDU</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>QUEROPT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>QUERPLN</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>QUERY</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>RELS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>RESAUTH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>ROLES</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>ROUTINA</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>ROUTINE</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>ROUTOPT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>ROUTSRC</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>SCHEMMA</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>SEQAUTH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>SEQDEP</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>SEQENC</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>STMT</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>STOGROU</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>STRINGS</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>SYNONYM</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>TABAUTH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>TABCNST</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>TABLEPA</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>TABLES</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>TABLESH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>TABLESP</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>TABPRTH</td>
<td>DYNAMIC, X</td>
</tr>
<tr>
<td>TABSTAH</td>
<td>DYNAMIC, X</td>
</tr>
</tbody>
</table>
Descriptions of the ACTDOPT options that are listed in Figure 175 follow. In some cases, the default value for the option is listed.

### ALLC=N

This option determines whether to display all panel titles, column heads, field prompts, and messages in uppercase characters (Y or N).

### AOPTS=ACMDOPD1

This option is no longer used to specify the ALTER or CHANGE MANAGER installation options module name to be used to run a CATALOG MANAGER worklist. Instead, CATALOG MANAGER uses the ACTvrDM plan to execute a worklist through the Execution component. The plan name is generated in the Execution JCL.

### AUDIT=Y

This option indicates whether to use audit logging (Y or N).

### BDSN='''&&HLQ..DBDBRM'''

This option indicates the DBRM library that CATALOG MANAGER uses when executing the BIND command.

---

**NOTE**

,R in the variable syntax indicates that the value specified will refresh the existing value of the variable in the user’s ISPF profile data set, if the time stamp of the installation options module is later than the time stamp in the user’s ISPF profile member.
Installation options

**BOPTS=ASUDOPD1**

This option indicates whether the DASD MANAGER PLUS product is also installed. If DASD MANAGER PLUS is installed, the installation options module name specified with this option is used to enable the use of the SPACE and STATS commands in CATALOG MANAGER.

---

**NOTE**

These parameters must match the load library and options module name that is used when installing DASD MANAGER PLUS.

This option is no longer used to run a CATALOG MANAGER worklist. Instead, CATALOG MANAGER uses the ACT\_vrDM plan to execute a worklist through the Execution component. The plan name is generated in the Execution JCL.

**BPLAN=ACT\_vrDB**

This option specifies the authorization plan for DSN commands. This plan is not used currently, but will be enabled in a future release.

**CATOP=Y**

This option indicates whether to perform the installation SYSADM check when CATALOG MANAGER is initialized (Y or N). Selecting Y starts a DB2 trace.

**CRS=N**

This option indicates whether issuing the SET PROFILE and SET PROFILE OFF commands requires SYSADM (System Administrator) authority (Y or N).

**CUP=Y**

This option specifies the conditional uppercase indicator.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Translate delimited identifiers to uppercase.</td>
</tr>
<tr>
<td>N</td>
<td>Do not translate delimited identifiers to uppercase.</td>
</tr>
</tbody>
</table>

**DBCS=N**

This option indicates if DB2 subsystem character strings can contain a mixture of SBCS and DBCS data or SBCS data only.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>SBCS and DBCS data</td>
</tr>
<tr>
<td>N</td>
<td>SBCS data only</td>
</tr>
</tbody>
</table>
**DPT=('.')**

This option indicates that the decimal point character for CATALOG MANAGER must be a comma or a period.

**DRO=O**

This option indicates the Drop Recovery option:

- **M** mandatory
- **O** optional
- **N** not used

**EPLAN=ACTvrDE**

This option specifies the name of the plan that enables access to the data editing and browsing functions.

**ESC='''**

This option indicates the SQL string delimiter. This delimiter must be an apostrophe (') or a quotation mark ("). You must select the one that matches the way your DB2 system was generated. The character that you do not select becomes the SQL escape character.

**GPLAN=ACTvrDG**

This option specifies the authorization plan for commands that will generate SQL for execution. This plan is not used currently, but will be enabled in a future release.

**GRPAT**

This option indicates the group attachment name for data sharing in a sysplex. This name is used as the SSID when JCL for utilities is generated.

**HDAL=N**

This option indicates whether to include aliases in the hierarchical describe (Y or N).

**HDIX=Y**

This option indicates whether to include indexes in the hierarchical describe (Y or N).

**HDPL=N**

This option indicates whether to include plans in the hierarchical describe (Y or N).
**HDSY=N**

This option indicates whether to include synonyms in the hierarchical describe (Y or N).

**HDTB=Y**

This option indicates whether to include tables in the hierarchical describe (Y or N).

**HDTR=Y**

This option indicates whether to include triggers in the hierarchical describe (Y or N).

**HDT5=Y**

This option indicates whether to include table spaces in the hierarchical describe (Y or N).

**HDVW=Y**

This option indicates whether to include views in the hierarchical describe (Y or N).

**HPLAN=ACT\vr\DH**

This option specifies the plan for displaying BMC utility status.

**ICCOL**

This option specifies the CATALOG MANAGER collection ID for indirect access.

**ICSYNC**

This option indicates the CATALOG MANAGER synonym creator ID that is used when installing catalog indirection for CATALOG MANAGER.

**JDSN=&&ZUSER..BMCCAT.JCL()**

This option specifies the default data set name that is used for utility JCL. The data set can be either a sequential or a partitioned data set. The product uses the utility name as the default member name automatically.

**KPLAN=ACT\vr\DK**

This option specifies the name of the DB2 commands plan.
**LDSN=&amp;ZUSER..BMCCAT.SQL**

This option specifies the name of the SQL output data set. CATALOG MANAGER dynamically allocates the data set the first time that it is used. If you want to preallocate this data set, use the following parameters:

\[
\text{DCB=(LRECL=4092,BKSIZE=4096,RECFM=VB)}
\]

**NOTE**

In many installations, allocation of data sets is controlled by user-written or third-party routines. If allocation fails, you should use alternate means, such as ISPF, to perform the allocations.

**LPLAN=ACTvrDL**

This option specifies the name of the CATALOG MANAGER logs maintenance plan.

**MAX=300**

This option indicates the maximum number of lines to generate in a list.

**MPLAN=ACTvrDM**

This option specifies the name of the CATALOG MANAGER main plan.

**PDSN=&amp;ZUSER..BMCCAT.PRINT**

This option specifies the name of the print output data set. CATALOG MANAGER dynamically allocates this data set the first time that it is used. If you want to preallocate this data set, use the following parameters:

\[
\text{DCB=(LRECL=4092,BKSIZE=27998,RECFM=VBA)}
\]

**NOTE**

In many installations, allocation of data sets is controlled by user-written or third-party routines. If allocation fails, you should use alternate means, such as ISPF, to perform the allocations.

**PLP=55**

This option indicates the number of print lines per page for the PRINT commands.
**POFDS='&&HLQ..UDBCNTL(&POFNAME)'**

This option specifies the name of the JCL Generation Product Options File (POF).

**RCCOL=ACT
vr
_D_MAIN**

This option specifies the CATALOG MANAGER collection ID for direct catalog access.

**SPLAN=ACT
vr
DS**

This option specifies the authorization plan for the SEARCH command. This plan is not used currently, but will be enabled in a future release.

**TDSN**

This option specifies the data set in which site utility profiles are saved. 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.

---

**NOTE**

Because the TDSN keyword is used for a site profile, the ,R (refresh) command will be ignored.

**TNCC=<->**

This parameter specifies the characters that replace the beginning and end of a truncated string in an object name that is too long to be displayed.

**TNLMR=M**

This parameter specifies the location of characters to be omitted in object names that are too long to be displayed:

- **L** Replace characters at the left end (beginning) of the name.
- **M** Replace characters in the middle of the name.
- **R** Replace characters at the right end (end) of the name.

**TRS=N**

This option indicates whether all users or just users with DB2 SYSADM authority can terminate utilities.
Y Only users with DB2 SYSADM authority can terminate utilities.
N Any user can terminate the utility.

**UCMD**

*(optional)* This option indicates the name of the user commands table. This user commands table contains your modifications to existing commands and any new commands, and overrides the primary commands table in member ACTCOMND in the HLQ.DBCNTL library. When you invoke CATALOG MANAGER, the product merges the primary commands table with your user commands table.

**UODSN=''&&ZUSER..BMCACT.USEROPT''**

This option specifies the name of the data set that contains values for user options in XML format. CATALOG MANAGER dynamically allocates the data set the first time that it is used. If you want to preallocate this data set, use the following parameters:

```
DCB=(LRECL=255,BLKSIZE=6124,RECFM=VB)
```

**NOTE**

In many installations, allocation of data sets is controlled by user-written or third-party routines. If allocation fails, you should use alternate means, such as ISPF, to perform the allocations.

**UPLAN=ACTvrDU**

This option specifies the name of the utilities plan.

**WDSN=&&ZUSER..BMCCAT.WORK**

This option specifies the name of the work data set in which statements that the HDDL command produces are stored. CATALOG MANAGER dynamically allocates the data set the first time that it is used.

If you want to preallocate this data set, use the following parameters:

```
DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
```

**NOTE**

In many installations, allocation of data sets is controlled by user-written or third-party routines. If allocation fails, you should use alternate means, such as ISPF, to perform the allocations.
This option specifies the name of the data set that defines the set of options for DESCRIBE package and DESCRIBE plan in XML format. The data set must be partitioned. ACTDESC is the default member name.

The $ACTSQLD options that are listed in Figure 175 indicate that CATALOG MANAGER uses dynamic SQL to access a catalog table. A user ID must have SELECT authority on the table to execute SQL on the table. DYNAMIC is the only valid value for these options. Table 63 lists the $ACTSQLD options and the corresponding table that CATALOG MANAGER accesses.

<table>
<thead>
<tr>
<th>Option</th>
<th>Catalog table</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUDPOL</td>
<td>SYSBM.SYSAUDITPOLICIES</td>
</tr>
<tr>
<td>AUTOALE</td>
<td>SYSBM.SYSAUTOALERTS</td>
</tr>
<tr>
<td>AUTORUN</td>
<td>SYSBM.SYSAUTORUN_HIST</td>
</tr>
<tr>
<td>AUTOWIN</td>
<td>SYSBM.SYSAUTOTIMEWINDOWS</td>
</tr>
<tr>
<td>AUXRELS</td>
<td>SYSBM.SYSAUXRELS</td>
</tr>
<tr>
<td>CHECKDE</td>
<td>SYSBM.SYSCHECKDEP</td>
</tr>
<tr>
<td>CHECKS</td>
<td>SYSBM.SYSCHECKS</td>
</tr>
<tr>
<td>CHECKS2</td>
<td>SYSBM.SYSCHECKS2</td>
</tr>
<tr>
<td>COLAUTH</td>
<td>SYSBM.SYSCOLAUTH</td>
</tr>
<tr>
<td>COLDISH</td>
<td>SYSBM.SYSCOLDIST_HIST</td>
</tr>
<tr>
<td>COLDISS</td>
<td>SYSBM.SYSCOLDISTSTATS</td>
</tr>
<tr>
<td>COLDIST</td>
<td>SYSBM.SYSCOLDIST</td>
</tr>
<tr>
<td>COLSTAT</td>
<td>SYSBM.SYSCOLSTATS</td>
</tr>
<tr>
<td>COLUMNNH</td>
<td>SYSBM.SYSCOLUMNNS_HIST</td>
</tr>
<tr>
<td>COLUMNS</td>
<td>SYSBM.SYSCOLUMNS</td>
</tr>
<tr>
<td>CONSTDE</td>
<td>SYSBM.SYSCONSTDEP</td>
</tr>
<tr>
<td>CTRL</td>
<td>SYSBM.SYSCONTROLS</td>
</tr>
<tr>
<td>CONTXT</td>
<td>SYSBM.SYSCONTEXT</td>
</tr>
<tr>
<td>COPY</td>
<td>SYSBM.SYSCOPY</td>
</tr>
<tr>
<td>CXATTR</td>
<td>SYSBM.SYSCXTXTTRUSTATTRS</td>
</tr>
<tr>
<td>CXAUTH</td>
<td>SYSBM.SYSCONTEXTAUTHIDS</td>
</tr>
<tr>
<td>DATABASES</td>
<td>SYSBM.SYSDATABASE</td>
</tr>
<tr>
<td>DATATYP</td>
<td>SYSBM.SYSDATATYPES</td>
</tr>
<tr>
<td>DBAUTH</td>
<td>SYSBM.SYSDBAUTH</td>
</tr>
<tr>
<td>DBRM</td>
<td>SYSBM.SYSDBRM</td>
</tr>
<tr>
<td>DEPEND</td>
<td>SYSBM.SYSDDEPENDENCIES</td>
</tr>
<tr>
<td>ENVIRON</td>
<td>SYSBM.SYSENVIRONMENT</td>
</tr>
<tr>
<td>FIELDS</td>
<td>SYSBM.SYSFIELDS</td>
</tr>
</tbody>
</table>
### Table 63  $ACTSQLD options (part 2 of 4)

<table>
<thead>
<tr>
<th>Option</th>
<th>Catalog table</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOREIGN</td>
<td>SYSIMB.SYSFOREIGNKEYS</td>
</tr>
<tr>
<td>INDEXES</td>
<td>SYSIMB.SYSINDEXES</td>
</tr>
<tr>
<td>INDEXH</td>
<td>SYSIMB.SYSINDEXES_HIST</td>
</tr>
<tr>
<td>INDEXEXPA</td>
<td>SYSIMB.SYSINDEXEXPART</td>
</tr>
<tr>
<td>INDEXEXPH</td>
<td>SYSIMB.SYSINDEXEXPART_HIST</td>
</tr>
<tr>
<td>INDEXEXH</td>
<td>SYSIMB.SYSINDEXSTATS_HIST</td>
</tr>
<tr>
<td>INDEXST</td>
<td>SYSIMB.SYSINDEXSTATS</td>
</tr>
<tr>
<td>IPLIST</td>
<td>SYSIMB.IPLIST</td>
</tr>
<tr>
<td>IPNAMES</td>
<td>SYSIMB.SYSIPNAMES</td>
</tr>
<tr>
<td>JARCONT</td>
<td>SYSIMB.SYSJARCONTENTS</td>
</tr>
<tr>
<td>JAROBJT</td>
<td>SYSIMB.SYSJAROBJECTS</td>
</tr>
<tr>
<td>JAVAPTH</td>
<td>SYSIMB.SYSVARIABLEAUTH</td>
</tr>
<tr>
<td>JAVOPTS</td>
<td>SYSIMB.SYSJAVAOPST</td>
</tr>
<tr>
<td>KCOLUSE</td>
<td>SYSIMB.SYSKEYCOLUSE</td>
</tr>
<tr>
<td>KEYS</td>
<td>SYSIMB.SYSKEYS</td>
</tr>
<tr>
<td>KEYTDST</td>
<td>SYSIMB.SYSKEYTGTDISTSTATS</td>
</tr>
<tr>
<td>KEYTG</td>
<td>SYSIMB.SYSKEYTGTDIST</td>
</tr>
<tr>
<td>KEYTGDH</td>
<td>SYSIMB.SYSKEYTGTDIST_HIST</td>
</tr>
<tr>
<td>KEYTGHS</td>
<td>SYSIMB.SYSKEYTARGETS_HIST</td>
</tr>
<tr>
<td>KEYTGST</td>
<td>SYSIMB.SYSKEYTARGETSTATS</td>
</tr>
<tr>
<td>KEYTGT</td>
<td>SYSIMB.SYSKEYTARGETS</td>
</tr>
<tr>
<td>LOBSTAT</td>
<td>SYSIMB.SYSLOBSTATS</td>
</tr>
<tr>
<td>LOBSTATH</td>
<td>SYSIMB.SYSLOBSTATS_HIST</td>
</tr>
<tr>
<td>LOCATIO</td>
<td>SYSIMB.LOCATIONS</td>
</tr>
<tr>
<td>LULIST</td>
<td>SYSIMB.LULIST</td>
</tr>
<tr>
<td>LUMODES</td>
<td>SYSIMB.LUMODES</td>
</tr>
<tr>
<td>LUNAMES</td>
<td>SYSIMB.LUNAMES</td>
</tr>
<tr>
<td>MODESEL</td>
<td>SYSIMB.MODESELECT</td>
</tr>
<tr>
<td>OBDS</td>
<td>SYSIMB.SYSOBDS</td>
</tr>
<tr>
<td>OBJROL</td>
<td>SYSIMB.SYSOBJROLEDEP</td>
</tr>
<tr>
<td>PACKAGE</td>
<td>SYSIMB.SYSPACKAGE</td>
</tr>
<tr>
<td>PACKAUT</td>
<td>SYSIMB.SYSPACKAUTH</td>
</tr>
<tr>
<td>PACKCPY</td>
<td>SYSIMB.SYSPACKCOPY</td>
</tr>
<tr>
<td>PACKDEP</td>
<td>SYSIMB.SYSPACKDEP</td>
</tr>
<tr>
<td>PACKLIS</td>
<td>SYSIMB.SYSPACKLIST</td>
</tr>
<tr>
<td>PACKSTM</td>
<td>SYSIMB.SYSPACKSTM</td>
</tr>
<tr>
<td>PARMS</td>
<td>SYSIMB.SYSPARMS</td>
</tr>
<tr>
<td>PENDDDL</td>
<td>SYSIMB.SYSPENDINGDDL</td>
</tr>
</tbody>
</table>
### Table 63  $ACTSQLD options (part 3 of 4)

<table>
<thead>
<tr>
<th>Option</th>
<th>Catalog table</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKSYSTE</td>
<td>SYSBM.SYSPKSYSTEM</td>
</tr>
<tr>
<td>PLAN</td>
<td>SYSBM.SYSPLAN</td>
</tr>
<tr>
<td>PLANAUT</td>
<td>SYSBM.SYSPLANAUTH</td>
</tr>
<tr>
<td>PLANDEP</td>
<td>SYSBM.SYSPLANDEP</td>
</tr>
<tr>
<td>PLSYTE</td>
<td>SYSBM.SYSPLSYSTEM</td>
</tr>
<tr>
<td>PROCEDU</td>
<td>SYSBM.SYSPROCEDURES</td>
</tr>
<tr>
<td>QUEROPT</td>
<td>SYSBM.SYSQUERYOPTS</td>
</tr>
<tr>
<td>QUERPLN</td>
<td>SYSBM.SYSQUERYPLAN</td>
</tr>
<tr>
<td>QUERY</td>
<td>SYSBM.SYSQUERY</td>
</tr>
<tr>
<td>RELS</td>
<td>SYSBM.SYSRELS</td>
</tr>
<tr>
<td>RESAUTH</td>
<td>SYSBM.SYSRESAUTH</td>
</tr>
<tr>
<td>ROLES</td>
<td>SYSBM.SYSROLES</td>
</tr>
<tr>
<td>ROUTINA</td>
<td>SYSBM.SYSROUTINEAUTH</td>
</tr>
<tr>
<td>ROUTINE</td>
<td>SYSBM.SYSROUTINES</td>
</tr>
<tr>
<td>ROUTOPT</td>
<td>SYSBM.SYSROUTINES_OPTS</td>
</tr>
<tr>
<td>ROUTSRC</td>
<td>SYSBM.SYSROUTINES_SRC</td>
</tr>
<tr>
<td>SCHEMAA</td>
<td>SYSBM.SYSSCHEMAAUTH</td>
</tr>
<tr>
<td>SEQAUTH</td>
<td>SYSBM.SYSSEQUENCEAUTH</td>
</tr>
<tr>
<td>SEQDEP</td>
<td>SYSBM.SYSSEQUENCESDEP</td>
</tr>
<tr>
<td>SEQUENC</td>
<td>SYSBM.SYSSEQUENCES</td>
</tr>
<tr>
<td>STMNT</td>
<td>SYSBM.SYSSMT</td>
</tr>
<tr>
<td>STOGROU</td>
<td>SYSBM.SYSSTOGROUP</td>
</tr>
<tr>
<td>STRINGS</td>
<td>SYSBM.SYSSTRINGS</td>
</tr>
<tr>
<td>SYNONYM</td>
<td>SYSBM.SYSSYNONYMS</td>
</tr>
<tr>
<td>TABAUTH</td>
<td>SYSBM.SYSTABAUTH</td>
</tr>
<tr>
<td>TABCNST</td>
<td>SYSBM.SYSTABCONST</td>
</tr>
<tr>
<td>TABLEPA</td>
<td>SYSBM.SYSTABLEPART</td>
</tr>
<tr>
<td>TABLES</td>
<td>SYSBM.SYSTABLES</td>
</tr>
<tr>
<td>TABLESH</td>
<td>SYSBM.SYSTABLES_HIST</td>
</tr>
<tr>
<td>TABLESP</td>
<td>SYSBM.SYSTABLESPACE</td>
</tr>
<tr>
<td>TABPRTH</td>
<td>SYSBM.SYSTABLEPART_HIST</td>
</tr>
<tr>
<td>TABSTAH</td>
<td>SYSBM.SYSTABSTATS_HIST</td>
</tr>
<tr>
<td>TABSTAT</td>
<td>SYSBM.SYSTABSTATS</td>
</tr>
<tr>
<td>TBLPROF</td>
<td>SYSBM.SYSTABLES_PROFILES</td>
</tr>
<tr>
<td>TRIGGER</td>
<td>SYSBM.SYSTRIGGERS</td>
</tr>
<tr>
<td>USERAUT</td>
<td>SYSBM.SYSUSERAUTH</td>
</tr>
<tr>
<td>USERNAM</td>
<td>SYSBM.USERNAMES</td>
</tr>
<tr>
<td>VIEWDEP</td>
<td>SYSBM.SYSVIEWDEP</td>
</tr>
<tr>
<td>Option</td>
<td>Catalog table</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>VIEWS</td>
<td>SYSIBM.SYSVIEWS</td>
</tr>
<tr>
<td>VOLUMES</td>
<td>SYSIBM.SYSVOLUMES</td>
</tr>
<tr>
<td>XMLREL</td>
<td>SYSIBM.SYSXMLRELS</td>
</tr>
<tr>
<td>XMLSTR</td>
<td>SYSIBM.SYSXMLSTRINGS</td>
</tr>
<tr>
<td>XSROBJ</td>
<td>SYSIBM.XSROBJECTS</td>
</tr>
</tbody>
</table>
JCL Generation product options

This appendix presents the following topics:

Overview ................................................................. 439
Product options .......................................................... 439

Overview

A keyword in the ALTER, CATALOG MANAGER, CHANGE MANAGER, and DASD MANAGER PLUS installation options modules, POFDS, specifies an 80-character sequential file. This file, the product options file (POF), contains keywords and values for the JCL Generation options. The file is located in the HLQ.UDBCNTL data set. When you install the products, only one POF is created. This POF, referred to as the initial POF, is initialized and populated with the default ISPF variables and values from the installation panels. Products that are installed at the same time share the initial POF.

For ALTER, CHANGE MANAGER, and DASD MANAGER PLUS, all of the POF keywords are included in the AJXPOFIN input stream (unless noted otherwise). These products use the keywords in the AJXPOFIN input stream in your initial POF and your user POF.

Product options

Figure 176 provides an example of a product options file.

Figure 176  Product options file (part 1 of 12)

POFDATE = 2012/11/13 09:20:37
*-------------------------------------------------------------
* POF WRITTEN FROM VERSION: V11.01.00
**Figure 176  Product options file (part 2 of 12)**

* FORMAT:
*   KEYWORD=PARM  COLUMNS 1-80.
*   PARM SYNTAX:
*     VALUE - EVERYTHING AFTER THE = IS CONSIDERED THE VALUE.
*     LEADING AND TRAILING BLANKS ARE REMOVED.
*     VALUE,(R) TO INDICATE REFRESH OPTION.
*     NO SPLITTING OF VALUE ACROSS LINES. IF IT WON'T FIT ON
*     LINE WITH KEYWORD, ENTER '>' AFTER = AND PUT THE PARM
*     ON NEXT LINE.
*     BLANK LINES ARE IGNORED.
*     ASTERISK IN COLUMN 1 INDICATES THAT LINE IS A COMMENT.

ACM_AMS = Y
ACM_BASDIAG = SYSOUT
ACM_BRPTDIAG = SYSOUT
ACM_BRPTDSN = '&PREFIX..BASELINE.REPORT'
ACM_CDLDSN = '&PREFIX..&SSID..CDL(CDL)'  
ACM_CDLPS = 15
ACM_CDLSS = 5
ACM_CDLU = SYSDA
ACM_CMPDIAG = SYSOUT
ACM_CPLCDLO = '&PREFIX..&SSID..CDL(CDL)'
ACM_CPLDIAG = SYSOUT
ACM_CPLWDSN = '&PREFIX..&SSID..&TASKID'
ACM_CPLWDSNO = '&PREFIX..&SSID..&WORKID'
ACM_DBRM1 =
ACM_DBRM2 =
ACM_DBRM3 =
ACM_DYNSORTW_NUM = 32
ACM_DYNSORTW_UNIT = SYSDA
ACM_GLID =
ACM_IBMR_MAP_REQ = Y
ACM_IMPDIAG = SYSOUT
ACM_JDSN = '&PREFIX..ANALYSIS(&WORKID)'
ACM_JDSNB = '&PREFIX..BASELINE(&WORKID)'
ACM_JDSNBG = '&PREFIX..JCLGEN(&WORKID)'
ACM_JDSNBR = '&PREFIX..BASELINE(BLRPTJCL)'
ACM_JDSNC = '&PREFIX..COMPARE(CMPJCL)'
ACM_JDSNCPL = '&PREFIX..TASKID(&TASKID)'
ACM_JDSNCPL0 = '&PREFIX..EXEC(&WORKID)'
ACM_JDSNE = '&PREFIX..EXEC(&WORKID)'
ACM_JDSNI = '&PREFIX..IMPORT(&WORKID)'
ACM_PARALLEL_MAXINIT = 3
ACM_PARALLEL_MININIT = 2
ACM_PARALLEL_WORKLST = N
ACM_PARALLEL_XIMGRP = XIMACM
ACM_PARALLEL_XIMPROC = XIMACM
ACM_PARALLEL_XIMSTRT = N
ACM_PARALLEL_XIMTRCE = N
ACM_PIC = N
ACM_SDSN = SYSOUT
ACM_SDSNE = SYSOUT
ACM_WDSN = '&PREFIX..&SSID..&WORKID'
ACM_WLODER =
ACM_WLODERMSG = Y
ACM_WLPS = 15
ACM_WLSS = 5
ACM_WLU = SYSDA
ADDLOAD1 = BMCRMD.V810INST.NONSMPE.LOAD,(R)
ADDLOAD2 = BMCRMD.V810INST.UDBLINK,(R)
ARCH_DATACLASS =
ARCH_DATACLASS_ALT =
ARCH_EXPDT =
ARCH_MGMTCLASS =
ARCH_MGMTCLASS_ALT =
ARCH_PREFIX = &PREFIX..&WKID
ARCH_PRIQTY = 10
ARCH_RETPD =
ARCH_SECQTY = 2
ARCH_STACK = N
ARCH_STORCLASS =
ARCH_STORCLASS_ALT =
ARCH_THRESH = 0
ARCH_UNIT = SYSDA
ARCH_UNIT_ALT =
ASU_XP_LOGD_DATAC=
ASU_XP_LOGD_MGMTC=
ASU_XP_LOGD_PRIQTY=10
ASU_XP_LOGD_SECQTY=2
ASU_XP_LOGD_STORC=
ASU_XP_LOGD_UNIT=SYSDA
ASU_XP_LOGDSN=&PREFIX..XPORT.LOG
ASU_XP_UIMSRVHOST=
ASU_XP_UIMSRVPORT=1
ASU_XP_UIMSRVTIMEOUT=300
BINDFAIL = N
BLRP_DATACLASS =
BLRP_DATACLASS_ALT =
BLRP_EXPDT =
BLRP_MGMTCLASS =
BLRP_MGMTCLASS_ALT =
BLRP_PREFIX = &PREFIX..&OBNOD
BLRP_PRIQTY = 10
BLRP_RETPD =
BLRP_SECQTY = 2
BLRP_STACK = N
BLRP_STORCLASS =
BLRP_STORCLASS_ALT =
BLRP_THRESH = 0
BLRP_UNIT = SYSDA
BLRP_UNIT_ALT =
BMC_CHECK_LOAD =
**Figure 176  Product options file (part 4 of 12)**

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_CHECK_OPTS</td>
<td>ACK$OPTS</td>
</tr>
<tr>
<td>BMC_COPY_LOAD</td>
<td>&amp;PREFIX.RNTM.BMCLINK.(R)</td>
</tr>
<tr>
<td>BMC_COPY_OPTS</td>
<td>ACP$OPTS</td>
</tr>
<tr>
<td>BMC_LOAD_LOAD</td>
<td>&amp;PREFIX.RNTM.BMCLINK.(R)</td>
</tr>
<tr>
<td>BMC_LOAD_OPTS</td>
<td>AMU$OPTS</td>
</tr>
<tr>
<td>BMC_RECOVER_LOAD</td>
<td>&amp;PREFIX.RNTM.BMCLINK.(R)</td>
</tr>
<tr>
<td>BMC_RECOVER_OPTS</td>
<td>AFR$OPTS</td>
</tr>
<tr>
<td>BMC_REORG_LOAD</td>
<td>&amp;PREFIX.RNTM.BMCLINK.(R)</td>
</tr>
<tr>
<td>BMC_REORG_OPTS</td>
<td>ARU$OPTS</td>
</tr>
<tr>
<td>BMC_REORG_XBMD =</td>
<td></td>
</tr>
<tr>
<td>BMC_UNLOAD_LOAD</td>
<td>&amp;PREFIX.RNTM.BMCLINK.(R)</td>
</tr>
<tr>
<td>BMC_UNLOAD_OPTS</td>
<td></td>
</tr>
<tr>
<td>CAT_LOAD</td>
<td>&amp;PREFIX.RNTM.BMCLINK.(R)</td>
</tr>
<tr>
<td>CHGMAN_LOAD</td>
<td>&amp;PREFIX.RNTM.BMCLINK.(R)</td>
</tr>
<tr>
<td>CLEANUP_RC</td>
<td>4</td>
</tr>
<tr>
<td>CNTL_DATACLASS</td>
<td></td>
</tr>
<tr>
<td>CNTL_EXPDT</td>
<td></td>
</tr>
<tr>
<td>CNTL_MGMTCLASS</td>
<td></td>
</tr>
<tr>
<td>CNTL_PREFIX</td>
<td>&amp;PREFIX..&amp;WKID..&amp;SSID</td>
</tr>
<tr>
<td>CNTL_PRIQTY</td>
<td>1</td>
</tr>
<tr>
<td>CNTL_RETPD</td>
<td></td>
</tr>
<tr>
<td>CNTL_SECQTY</td>
<td>1</td>
</tr>
<tr>
<td>CNTL_STORCLASS</td>
<td></td>
</tr>
<tr>
<td>CNTL_UNIT</td>
<td>SYSDA</td>
</tr>
<tr>
<td>CNTLMOUT_DSN</td>
<td>&amp;PREFIX..&amp;SSID..CNTLMOUT(&amp;JOBNAME)</td>
</tr>
<tr>
<td>CNTLMOUTSCH_DSN</td>
<td>&amp;PREFIX..&amp;SSID..CNTLMOUTH(&amp;JOBNAME)</td>
</tr>
<tr>
<td>CPYEXP_DATACLASS</td>
<td></td>
</tr>
<tr>
<td>CPYEXP_EXPDT</td>
<td></td>
</tr>
<tr>
<td>CPYEXP_MGMTCLASS</td>
<td></td>
</tr>
<tr>
<td>CPYEXP_PREFIX</td>
<td>&amp;PREFIX..&amp;WKID</td>
</tr>
<tr>
<td>CPYEXP_RETPD</td>
<td></td>
</tr>
<tr>
<td>CPYEXP_STORCLASS</td>
<td></td>
</tr>
<tr>
<td>CPYEXP_SUPPRESS_SUFF</td>
<td>N</td>
</tr>
<tr>
<td>CPYEXP_UNIT</td>
<td>SYSDA</td>
</tr>
<tr>
<td>DASD_LOAD</td>
<td>&amp;PREFIX.DBLINK.(R)</td>
</tr>
<tr>
<td>DASDDOPT</td>
<td>RASUBFD.(R)</td>
</tr>
<tr>
<td>DATA_PACKER_LOAD</td>
<td>&amp;PREFIX.RNTM.BMCLINK.(R)</td>
</tr>
<tr>
<td>DATASETSIZING</td>
<td>N</td>
</tr>
<tr>
<td>DATAWK_NBR</td>
<td>4</td>
</tr>
<tr>
<td>DATAWK_UNIT</td>
<td>SYSDA</td>
</tr>
<tr>
<td>DB2EXIT</td>
<td>SYS3.DEBF.DSNEXIT.(R)</td>
</tr>
<tr>
<td>DB2LOAD</td>
<td>CSGI.DB2V81M.DSNLOAD.(R)</td>
</tr>
<tr>
<td>DEF_GDG_BASE</td>
<td>N</td>
</tr>
<tr>
<td>DEF_GDG_LIMIT</td>
<td>10</td>
</tr>
<tr>
<td>DEF_GDG_NOSCR</td>
<td>N</td>
</tr>
<tr>
<td>DEF_GDG2_LIMIT</td>
<td>10</td>
</tr>
<tr>
<td>DIAG_MSGCLASS</td>
<td></td>
</tr>
<tr>
<td>DISC_DATACLASS</td>
<td></td>
</tr>
<tr>
<td>DISC_DATACLASS_ALT</td>
<td></td>
</tr>
<tr>
<td>DISC_EXPDT</td>
<td></td>
</tr>
<tr>
<td>DISC_MGMTCLASS</td>
<td></td>
</tr>
</tbody>
</table>
DISC_MGMTCLASS_ALT =
DISC_PREFIX = &PREFIX..&WORKID..&OBNOD
DISC_PRIQTY = 10
DISC_RETPOD =
DISC_SECQTY = 2
DISC_STORCLASS =
DISC_STORCLASS_ALT =
DISC_THRESH = 0
DISC_UNIT = SYSDA
DISC_UNIT_ALT =
DISP_ALLOW_POPUP = Y
DISP_AUTO_TAB = +
DISP_LOCATION = M
DISP_OMIT_CHAR = <>
DISP_STATS = N
DISP_VAR_DBG = N
DROP_NOIC = N
DSNCHECK44 = N
DSNTIAD_PLAN =
ERR_DATACLASS =
ERR_DATACLASS_ALT =
ERR_EXPDT =
ERR_MGMTCLASS =
ERR_MGMTCLASS_ALT =
ERR_PREFIX = &PREFIX..&WKID..&STPN
ERR_PRIQTY = 10
ERR_RETPOD =
ERR_SECQTY = 2
ERR_STORCLASS =
ERR_STORCLASS_ALT =
ERR_THRESH = 0
ERR_UNIT = SYSDA
ERR_UNIT_ALT =
EXEC_LOAD = &PREFIX.RNTM.BMCLINK,(R)
FCPY_DATACLASS =
FCPY_EXPDT =
FCPY_MGMTCLASS =
FCPY_MGMTCLASS_ALT =
FCPY_PREFIX = &PREFIX..&OBNOD..P&PART
FCPY_PRIQTY = 10
FCPY_RETPOD =
FCPY_SECQTY = 2
FCPY_STORCLASS =
FCPY_SUPPRESS_SUFF = N
FCPY_UNIT = SYSDA
FILT_DATACLASS =
FILT_EXPDT =
FILT_MGMTCLASS =
FILT_PREFIX = &PREFIX..&WKID..&STPN
FILT_PRIQTY = 10
FILT_RETPOD =
FILT_SECQTY = 2
FILT_STORCLASS = 
FILT_UNIT = SYSDA
GDG_MODEL = SYS1.MODEL
HASHFAIL = N
HASHWARNR =
INCLUDE_SYSPRIN2 = N
IOALOAD1 = DFD.V6218.IOAI.LOAD
IOALOAD2 = DFD.V6218.IOAI.CTRANS
JCLCLEANUP = N
JCLLIB =
JES3 = N
JOB_INCUDE_MEMBER =
JOBCARD1 = //&USERID.&JOBCHAR JOB (&ZACCTNUM),+'&PGMR'.
JOBCARD2 = // CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),
JOBCARD3 = // NOTIFY=&USERID
JOBCARD4 = /*ROUTE XEQ BMCPLX1
JOBCARD5 = /*JOBPARM SYSAFF=&ZSYSID
LISTDEF_DSN =
LL_CLIB = CLIB
LL_CLIB2 =
LL_CLIB3 =
LL_CLIB4 =
LL_CLIB5 =
LL_LINK = LINK
LL_LINK2 =
LL_LINK3 =
LL_LINK4 =
LL_LINK5 =
LL_MLIB = MLIB
LL_MLIB2 =
LL_MLIB3 =
LL_MLIB4 =
LL_MLIB5 =
LL_PLIB = PLIB
LL_PLIB2 =
LL_PLIB3 =
LL_PLIB4 =
LL_PLIB5 =
LL_SLIB = SLIB
LL_SLIB2 =
LL_SLIB3 =
LL_SLIB4 =
LL_SLIB5 =
LL_TLIB = TLIB
LL_TLIB2 =
LL_TLIB3 =
LL_TLIB4 =
LL_TLIB5 =
LL_XML = XML
LL_XML2 =
LL_XML3 =

Figure 176 Product options file (part 6 of 12)
**Figure 176  Product options file (part 7 of 12)**

```plaintext
LL.XML4 =
LL.XML5 =
LLQ = BMC,(R)
LOGWK_NBR = 4
LOGWK_UNIT = SYSDA
MAP_DATACLASS =
MAP_DATACLASS_ALT =
MAP_EXPDT =
MAP_MGMTCLASS =
MAP_MGMTCLASS_ALT =
MAP_PREFIX = &PREFIX..&WKID..&SSID
MAP_PRIQTY = 10
MAP_RETPD =
MAP_SECQTY = 2
MAP_STORCLASS =
MAP_STORCLASS_ALT =
MAP_THRESH = 0
MAP_UNIT = SYSDA
MAP_UNIT_ALT =
MAX_CYL = 99999
MAX_PRIQTY = 2000
MAX_SECQTY = 200
MAX_UNITCNT =
MEMLIMIT =
ORTPARM_DSN =
PCPY1_DATACLASS =
PCPY1_DATACLASS_ALT =
PCPY1_EXPDT =
PCPY1_MGMTCLASS =
PCPY1_MGMTCLASS_ALT =
PCPY1_PREFIX = &PREFIX..&OBNOD..P&PART
PCPY1_PRIQTY = 10
PCPY1_RETPD =
PCPY1_SECQTY = 2
PCPY1_STACK = N
PCPY1_STORCLASS =
PCPY1_STORCLASS_ALT =
PCPY1_SUPPRESS_SUFF = N
PCPY1_THRESH = 0
PCPY1_UNIT = SYSDA
PCPY1_UNIT_ALT =
PCPY2_DATACLASS =
PCPY2_DATACLASS_ALT =
PCPY2_EXPDT =
PCPY2_MGMTCLASS =
PCPY2_MGMTCLASS_ALT =
PCPY2_PREFIX = &PREFIX..&OBNOD..P&PART
PCPY2_PRIQTY = 10
PCPY2_RETPD =
PCPY2_SECQTY = 2
PCPY2_STACK = N
```
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCPY2_STORCLASS</td>
<td></td>
</tr>
<tr>
<td>PCPY2_STORCLASS_ALT</td>
<td></td>
</tr>
<tr>
<td>PCPY2_SUPPRESS_SUFF</td>
<td>N</td>
</tr>
<tr>
<td>PCPY2_THRESH</td>
<td>0</td>
</tr>
<tr>
<td>PCPY2_UNIT</td>
<td>SYSDA</td>
</tr>
<tr>
<td>PCPY2_UNIT_ALT</td>
<td></td>
</tr>
<tr>
<td>PRE_JOBS tây INCLUDE</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCHECK_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCHECK_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCCOPY_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCCOPY_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCPPRS_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCPPRS_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_BMLOAD_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMLOAD_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCRECOVER_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCRECOVER_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCREORG_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCREORG_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCSTATS_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCSTATS_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCSTOP_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCSTOP_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCCTRIG_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCCTRIG_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCUNLOAD_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCUNLOAD_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCUPRS_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_BMCUPRS_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_DSNUTILB_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_DSNUTILB_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_DSNCOPY_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_DSNCOPY_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_GEN_SET_VAR</td>
<td>N</td>
</tr>
<tr>
<td>PROC_IDCAMS_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_IDCAMS_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_IEFBR14_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_IEFBR14_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_TSO_NAME</td>
<td></td>
</tr>
<tr>
<td>PROC_TSO_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_USE</td>
<td>N</td>
</tr>
<tr>
<td>PROC_USER_LINE_STEP</td>
<td></td>
</tr>
<tr>
<td>PROC_USER_DEFINED</td>
<td></td>
</tr>
<tr>
<td>PUNCH_DATACLASS</td>
<td></td>
</tr>
<tr>
<td>PUNCH_EXPDT</td>
<td></td>
</tr>
<tr>
<td>PUNCH_MGMTCLASS</td>
<td></td>
</tr>
<tr>
<td>PUNCH_PREFIX</td>
<td>&amp;PREFIX..&amp;WKID..&amp;STEPN</td>
</tr>
<tr>
<td>PUNCH_PRIQTY</td>
<td>1</td>
</tr>
<tr>
<td>PUNCH_RETPD</td>
<td>1</td>
</tr>
<tr>
<td>PUNCH_SECQTY</td>
<td>1</td>
</tr>
<tr>
<td>PUNCH_STORCLASS</td>
<td></td>
</tr>
</tbody>
</table>
Figure 176  Product options file (part 9 of 12)

```
PUNCH_UNIT = SYSDA
RCPY1_DATACLASS =
RCPY1_DATACLASS_ALT =
RCPY1_EXPDT =
RCPY1_MGMTCLASS =
RCPY1_MGMTCLASS_ALT =
RCPY1_PREFIX = &PREFIX..&OBNOD..P&PART
RCPY1_PRIQTY = 10
RCPY1_RETPD =
RCPY1_SECQTY = 2
RCPY1_STACK = N
RCPY1_STORCLASS =
RCPY1_STORCLASS_ALT =
RCPY1_SUPPRESS_SUFF = N
RCPY1_THRESH = 0
RCPY1_UNIT = SYSDA
RCPY1_UNIT_ALT =
RCPY2_DATACLASS =
RCPY2_DATACLASS_ALT =
RCPY2_EXPDT =
RCPY2_MGMTCLASS =
RCPY2_MGMTCLASS_ALT =
RCPY2_PREFIX = &PREFIX..&OBNOD..P&PART
RCPY2_PRIQTY = 10
RCPY2_RETPD =
RCPY2_SECQTY = 2
RCPY2_STACK = N
RCPY2_STORCLASS =
RCPY2_STORCLASS_ALT =
RCPY2_SUPPRESS_SUFF = N
RCPY2_THRESH = 0
RCPY2_UNIT = SYSDA
RCPY2_UNIT_ALT =
REBINDFAIL = N
REBINDRC =
REGION = 0M
REORG_MAPTAB =
REPT_DATACLASS =
REPT_DATACLASS_ALT =
REPT_EXPDT =
REPT_MGMTCLASS =
REPT_MGMTCLASS_ALT =
REPT_PREFIX = &PREFIX..&WKID
REPT_PRIQTY = 10
REPT_RETPD =
REPT_SECQTY = 2
REPT_STORCLASS =
REPT_STORCLASS_ALT =
REPT_THRESH = 0
REPT_UNIT = SYSDA
REPT_UNIT_ALT =
```
RUNTIME_HLQ = &PREFIX.RNTM,(R)
SCHED_TRIG_CTLM_JOBS=N
SORTWK_NBR = 4
SORTWK_PRIQTY = 10
SORTWK_SECQTY = 2
SORTWK_UNIT = SYSDA
SOLEXP_LOAD = &PREFIX.LOAD,(R)
SRTOUT_DATACLASS =
SRTOUT_DATACLASS_ALT =
SRTOUT_EXPDT =
SRTOUT_MGMTCLASS =
SRTOUT_MGMTCLASS_ALT =
SRTOUT_PREFIX = &PREFIX..&WKID..&STEPN
SRTOUT_PRIQTY = 10
SRTOUT_RETPD =
SRTOUT_SECQTY = 2
SRTOUT_STORCLASS =
SRTOUT_STORCLASS_ALT =
SRTOUT_THRESH = 0
SRTOUT_UNIT = SYSDA
SRTOUT_UNIT_ALT =
STEP_INCLUDE_MEMBER =
STOPWAIT = 3
STOPWTSECS = 10
SUPPRESS_COMMENTS = N
SYNCDDELETE = N
SYSEXEC = &PREFIX.RNTM.BMCREXX,(R)
SYSTEM_MLIB = SYS1.PROD.ISPMENU,(R)
SYSUT_DATACLASS =
SYSUT_DATACLASS_ALT =
SYSUT_EXPDT =
SYSUT_MGMTCLASS =
SYSUT_MGMTCLASS_ALT =
SYSUT_PREFIX = &PREFIX..&WKID..&STEPN
SYSUT_PRIQTY = 10
SYSUT_RETPD =
SYSUT_SECQTY = 2
SYSUT_STORCLASS =
SYSUT_STORCLASS_ALT =
SYSUT_THRESH = 0
SYSUT_UNIT = SYSDA
SYSUT_UNIT_ALT =
SZDEV = 3390
TAPE_EXPDT =
TAPE_RETPD =
TAPE_VOLCNT = 99
TAPE1 = CART
TAPE2 = TAPE
TAPE3 = TAPE
TEMPLATE_DSN =
TEMPUNIT = SYSDA
TIMEPARM =
TRTCH =
TSOPROGRAM =
TSOSUBEXIT = N
ULLQ = ,(R)
UNLD_FREF_DATACLASS =
UNLD_FREF_DIRBLOCK = 250
UNLD_FREF_MGMTCLASS =
UNLD_FREF_PREFNAME = &PREFIX..&MSSID..&WORKID8
UNLD_FREF_PRIQTY = 10
UNLD_FREF_SECQTY = 2
UNLD_FREF_STORCLASS =
UNLD_FREF_SUPPR_SUFF = N
UNLD_FREF_UNIT = SYSDA
UNLD1_DATACLASS =
UNLD1_DATACLASS_ALT =
UNLD1_EXPDT =
UNLD1_MGMTCLASS =
UNLD1_MGMTCLASS_ALT =
UNLD1_PREFNAME = &PREFIX..&MSSID..&WORKID8
UNLD1_PRIQTY = 10
UNLD1_RETPD =
UNLD1_SECQTY = 2
UNLD1_STACK = N
UNLD1_STORCLASS =
UNLD1_STORCLASS_ALT =
UNLD1_SUPPRESS_SUFF = N
UNLD1_THRESH = 0
UNLD1_UNIT = SYSDA
UNLD1_UNIT_ALT =
UNLD2_DATACLASS =
UNLD2_DATACLASS_ALT =
UNLD2_EXPDT =
UNLD2_MGMTCLASS =
UNLD2_MGMTCLASS_ALT =
UNLD2_PREFNAME = &PREFIX..&MSSID..&WORKID8
UNLD2_PRIQTY = 10
UNLD2_RETPD =
UNLD2_SECQTY = 2
UNLD2_STACK = N
UNLD2_STORCLASS =
UNLD2_STORCLASS_ALT =
UNLD2_SUPPRESS_SUFF = N
UNLD2_THRESH = 0
UNLD2_UNIT = SYSDA
UNLD2_UNIT_ALT =
UNLD3_DATACLASS =
UNLD3_EXPDT =
UNLD3_MGMTCLASS =
UNLD3_PREFNAME = &PREFIX..&MSSID..&WORKID8
UNLD3_RETPD =
Descriptions of the keywords that are listed in Figure 176 on page 439 follow.

**2MEGSQ=N**

This keyword indicates whether to allocate a 2-MB buffer for large SQL statements.

This keyword is not included in the AJXPOFIN input stream.

**ACM_AMS=Y**

For ALTER and CHANGE MANAGER, this keyword controls whether Analysis, by default, generates AMS statements (IDCAMS DELETE and DEFINE) in the worklist. The following values are valid:

- **Y** generates AMS statements (IDCAMS DELETE and DEFINE) in a worklist
- **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

You can use the INCLUDE (AMS) keyword to override this value.

This keyword is not included in the AJXPOFIN input stream.
**ACM_BASDIAG=SYSOUT**

For CHANGE MANAGER, this keyword sets the default value for the Baseline diagnostic output data set name.

This keyword is not included in the AJXPOFIN input stream.

**ACM_BRPTDIAG=SYSOUT**

For CHANGE MANAGER, this keyword specifies the default name for the Baseline Report diagnostic output data set.

This keyword is not included in the AJXPOFIN input stream.

**ACM_BRPTDSN='&PREFIX..BASELINE.REPORT’**

For CHANGE MANAGER, this keyword specifies the default name for the Baseline Report data set name.

This keyword is not included in the AJXPOFIN input stream.

**ACM_CDLDSN='&PREFIX..&SSID..CDL(CDL)'**

For CHANGE MANAGER, this keyword defines the default data set name for generated Change Definition Language (CDL) statements.

This keyword is not included in the AJXPOFIN input stream.

**ACM_CDLPS=15**

For CHANGE MANAGER, this keyword 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.

**ACM_CDLSS=5**

For CHANGE MANAGER, this keyword 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.

**ACM_CDLU=SYSDA**

For CHANGE MANAGER, this keyword defines the default unit for the CDL data set.
Product options

This keyword is not included in the AJXPOFIN input stream.

**ACM_CMPDIAG=SYSOUT**

For CHANGE MANAGER, this keyword defines the default value for the Compare diagnostic output data set.

This keyword is not included in the AJXPOFIN input stream.

**ACM_CPLCDLO='&PREFIX..&SSID..CDL(CDL)’**

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.

**ACM_CPLDIAG=SYSOUT**

For CHANGE MANAGER, this keyword specifies the default name for the CM/PILOT component’s diagnostic output data set.

This keyword is not included in the AJXPOFIN input stream.

**ACM_CPLWDSN='&PREFIX..&SSID..&TASKID’**

For CHANGE MANAGER, this keyword 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.

**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.

**ACM_DBRM1**
**ACM_DBRM2**
**ACM_DBRM3**

For ALTER and CHANGE MANAGER, these keywords specify the name of a default DBRM library.
This keyword is not included in the AJXPOFIN input stream.

**ACM_DYNSORTW_NUM = 32**

For ALTER and CHANGE MANAGER, this keyword specifies the number of dynamically allocated sortwork data sets that the BMC REORG PLUS or IBM REORG utility uses.

**ACM_DYNSORTW_UNIT = SYSDA**

For ALTER and CHANGE MANAGER, this keyword specifies the unit for dynamically allocated sortwork data sets. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**ACM_GLID = id**

For ALTER and CHANGE MANAGER, this keyword 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.

**ACM_IBMR_MAP_REQ=Y**

For ALTER and CHANGE MANAGER, this keyword indicates whether to include the name of the mapping table in the syntax for the IBM REORG utility. The IBM REORG utility uses the mapping table to map the row IDs (RIDs) in the source table the RIDs in the target table.

**NOTE**

The REORG PLUS utility invokes the IBM DSNUTILB utility control program to enable certain features. If you have specified to use the REORG PLUS utility, you still need to specify mapping table information. For information about the features for which REORG PLUS invokes DSNUTILB, see the *REORG PLUS for DB2 Reference Manual*.

**ACM_IMPDIAG=SYSOUT**

For ALTER and CHANGE MANAGER, this keyword defines the default name for the Import diagnostic output data set.

This keyword is not included in the AJXPOFIN input stream.
**ACM_JDSN=’&PREFIX..ANALYSIS(&WORKID)’**

For ALTER and CHANGE MANAGER, this keyword 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.

**ACM_JDSNB=’&PREFIX..BASELINE(&WORKID)’**

For CHANGE MANAGER, this keyword 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.

**ACM_JDSNBG=’&PREFIX..JCLGEN(&WORKID)’**

For ALTER and CHANGE MANAGER, this keyword defines the default data set name that is used for batch 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.

**ACM_JDSNBR=’&PREFIX..BASELINE(BLRPTJCL)’**

For CHANGE MANAGER, this keyword 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.

**ACM_JDSNC=’&PREFIX..COMPARE(CMPJCL)’**

For CHANGE MANAGER, this keyword 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.
**ACM_JDSNCPL=’&PREFIX..TASKID(&TASKID)’**

For CHANGE MANAGER, this keyword 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.

**ACM_JDSNCPLO=’&PREFIX..EXEC(&WORKID)’**

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.

**ACM_JDSNE=’&PREFIX..EXEC(&WORKID)’**

For ALTER and 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 products automatically use the work ID as the member name.

This keyword is not included in the AJXPOFIN input stream.

**ACM_JDSNI=’&PREFIX..IMPORT(&WORKID)’**

For ALTER and CHANGE MANAGER, this keyword 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.

**ACM_PARALLEL_MAXINIT=3**

For the Database Administration solution, this keyword specifies the maximum number of the 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.
ACM_PARALLEL_MININIT=2

For the Database Administration solution, this keyword specifies the minimum number of the XIM initiators to use when executing a worklist in parallel. If the minimum number of XIM initiators is not available, the worklist does not run. The valid range of values is 1 through 8.

ACM_PARALLEL_WORKLST=N

For the Database Administration solution, this keyword indicates whether a CHANGE MANAGER worklist should be executed in parallel.

Y  Execute the worklist in parallel.

However, if adequate XIM 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  Execute the worklist sequentially, even if the required parallelism worklist commands are included in the worklist.

ACM_PARALLEL_XIMGRP=XIMACM

For the Database Administration solution, this keyword 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.

ACM_PARALLEL_XIMPROC=XIMACM

For the Database Administration solution, this keyword 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.

ACM_PARALLEL_XIMSTRT=N

For the Database Administration solution, this keyword indicates whether the XIM technology should be started automatically.
**ACM_PARALLEL_XIMTRCE=N**

For the Database Administration solution, this keyword indicates whether tracing is used during the execution of a worklist.

- **Y** Write tracing records to the AEXPTRAC output data set.
  
  AEXPTRAC is dynamically allocated and the output is written to SYSOUT.

- **N** Do not use tracing, even if an //AEXPTRAC DD statement is specified in the JCL.

**ACM_PIC=N**

For ALTER and CHANGE MANAGER, this keyword 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.

**ACM_SDSN=SYSOUT**

For ALTER and CHANGE MANAGER, this keyword 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.

**ACM_SDSNE=SYSOUT**

For ALTER and CHANGE MANAGER, this keyword 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.

**ACM_WDSN='&PREFIX..&SSID..&WORKID'**

For ALTER and CHANGE MANAGER, this keyword defines the default data set name for a worklist that Analysis generates.

This keyword is not included in the AJXPOFIN input stream.
**ACM_WLORDER**

For ALTER and CHANGE MANAGER, this keyword specifies how the Analysis component sorts objects in a worklist.

- **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 by 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.

**ACM_WLORDERMSG=Y**

For ALTER and CHANGE MANAGER, this keyword 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.

**ACM_WLPS=15**

For ALTER and CHANGE MANAGER, this keyword defines, in tracks, the default primary space allocation for the worklist.

This keyword is not included in the AJXPOFIN input stream.

**ACM_WLSS=5**

For ALTER and CHANGE MANAGER, this keyword defines, in tracks, the default secondary space allocation for the worklist.

This keyword is not included in the AJXPOFIN input stream.

**ACM_WLU=SYSDA**

For ALTER and CHANGE MANAGER, this keyword defines the default worklist unit.

This keyword is not included in the AJXPOFIN input stream.
ADDLOAD1

This keyword defines the additional LINK library.

_TIP_

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

ADDLOAD2=HLQUDBLINK

This keyword defines the override LINK library. This LINK library is placed first in any STEPLIB concatenation.

_TIP_

If you are using a runtime environment, you can indicate the data set name for a different SSID by appending the &SSID or &MSSID symbolic variable to the name.

ARCH_DATACLASS

This keyword specifies the IBM Storage Management Subsystem (SMS) definition for the data class associated with the archive data set.

ARCH_DATACLASS_ALT

This keyword specifies the SMS definition for the data class associated with the archive data set (used if the threshold is exceeded).

ARCH_EXPDT

This keyword specifies the expiration date of the archive data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyyyddd or yyyy/ddd.

ARCH_MGMTCLASS

This keyword specifies the SMS definition for the storage class associated with the archive data set.

ARCH_MGMTCLASS_ALT

This keyword specifies the SMS definition for the storage class associated with the archive data set (used if the threshold is exceeded).
**ARCH_PREFIX=&PREFIX..&WKID**

This keyword specifies the prefix for the name of the archive discard data set.

**ARCH_PRIQTY=10**

This keyword specifies the primary allocation (in cylinders) for the archive discard data set if `DATASETSIZING=N` or if an error in sizing occurs. Valid values are 1 through 99999.

**ARCH_RETPD**

This keyword specifies the retention period for the archive data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

**ARCH_SECQTY=2**

This keyword specifies the secondary allocation (in cylinders) for the archive discard data set if `DATASETSIZING=N` or if an error in sizing occurs. Valid values are 1 through 99999.

**ARCH_STACK=N**

This keyword specifies whether to stack the archive data set on a tape with data sets of the same type (Y or N).

**ARCH_STORCLASS**

This keyword displays the SMS definition for the storage class associated with the archive data set.

**ARCH_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the archive data set (used if the threshold is exceeded).

**ARCH_THRESH=0**

This keyword 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.
**ARCH_UNIT=SYSDA**

This keyword specifies the unit for the archive discard data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**ARCH_UNIT_ALT**

This keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**ASU_XP_LOGD_DATACLASS=**

For DASD MANAGER PLUS, this keyword specifies the SMS data class and the allocation attributes of the Export log file.

**ASU_XP_LOGD_MGMTCLASS=**

For DASD MANAGER PLUS, this keyword specifies the SMS management class that defines the migration, retention, and backup requirements of the Export log file.

**ASU_XP_LOGD_PRIQTY=10**

For DASD MANAGER PLUS, this keyword specifies the primary allocation for the Export log file.

**ASU_XP_LOGD_SECIQTY=2**

For DASD MANAGER PLUS, this keyword specifies the secondary allocation for the Export log file.

**ASU_XP_LOGD_STORCLASS=10**

For DASD MANAGER PLUS, this keyword specifies the SMS storage class that defines the processing requirements of the Export log file.

**ASU_XP_LOGD_UNIT=SYSDA**

For DASD MANAGER PLUS, this keyword specifies the unit for the Export log file.

**ASU_XP_LOGD_LOGDSN=&PREFIX..XPORT.LOG**

For DASD MANAGER PLUS, this keyword specifies the Export log file.
**ASU_XP_UIMSRVHOST=**

For DASD MANAGER PLUS, this keyword specifies the host name of the primary UIM server which contains the host definitions repository for the Export utility.

**ASU_XP_UIMSRVPORT=1**

For DASD MANAGER PLUS, this keyword specifies the port number of the primary UIM server that contains the host definitions repository for the Export utility.

**ASU_XP_UIMSRVTIMEOUT=300**

For DASD MANAGER PLUS, this keyword specifies the UIM timeout parameter that determines how long the Export utility should wait for a response from the UIM server before timing out.

**BINDFAIL=N**

This keyword 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.

**BLRP_DATACLASS**

For CHANGE MANAGER, this keyword specifies the SMS definition for the data class associated with the baseline recovery point data set.

**BLRP_DATACLASS_ALT**

For CHANGE MANAGER, this keyword specifies the SMS definition for the data class associated with the baseline recovery point data set (used if the threshold is exceeded).

**BLRP_EXPDT**

For CHANGE MANAGER, this keyword specifies the expiration date of the baseline recovery point data set on tape. The valid values are *yyddd* or *yyyy/ddd*.

**BLRP_MGMTCLASS**

For CHANGE MANAGER, this keyword specifies the SMS definition for the storage class associated with the baseline recovery point data set.
**BLRP_MGMTCLASS_ALT**

For CHANGE MANAGER, this keyword specifies the SMS definition for the storage class associated with the baseline recovery point data set (used if the threshold is exceeded).

**BLRP_PREFIX=&PREFIX..&OBNOD**

For CHANGE MANAGER, this keyword defines the high-level qualifier, or prefix, used for data sets containing data stored for a baseline recovery point.

**BLRP_PRIQTY=10**

For CHANGE MANAGER, this keyword defines the primary allocation quantity for baseline recovery point data sets if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**BLRP_RETPD**

For CHANGE MANAGER, this keyword specifies the retention period for the baseline recovery point data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

**BLRP_SECQTY=2**

For CHANGE MANAGER, this keyword defines the secondary allocation quantity for baseline recovery point data sets if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 to 99999.

**BLRP_STACK=N**

For CHANGE MANAGER, this keyword specifies whether to stack full-recovery baseline data sets on tape (Y or N).

**BLRP_STORCLASS**

For CHANGE MANAGER, this keyword specifies the SMS definition for the storage class associated with the baseline recovery point data set.

**BLRP_STORCLASS_ALT**

For CHANGE MANAGER, this keyword specifies the SMS definition for the storage class associated with the baseline recovery point data set (used if the threshold is exceeded).
**BLRP_THRESH=0**

For CHANGE MANAGER, this keyword specifies the maximum anticipated size for the baseline recovery point 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.

**BLRP_UNIT=SYSDA**

For CHANGE MANAGER, this keyword specifies the unit used for the baseline recovery point data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**BLRP_UNIT_ALT**

For CHANGE MANAGER, this keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**BMC_CHECK_LOAD**

This keyword 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.

---

**TIP**

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

**BMC_CHECK_OPTS=ACKSOPTS**

This keyword 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.

**BMC_COPY_LOAD**

This keyword 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.
BMC_COPY_OPTS=ACP$OPTS

This keyword 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.

BMC_LOAD_LOAD

This keyword 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.

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

BMC_LOAD_OPTS=AMU$OPTS

This keyword specifies the name of the BMC LOADPLUS utility installation options module. 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.

BMC_RECOVER_LOAD

This keyword specifies the name of the LINK library for the BMC RECOVER PLUS utility. 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.

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

This keyword specifies the name of the installation options module for the BMC RECOVER PLUS utility. 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.

**BMC_REORG_LOAD**

This keyword specifies the name of the LINK library for the BMC REORG PLUS utility. 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.

**TIP**

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

**BMC_REORG_OPTS=ARU$OPTS**

This keyword specifies the name of the installation options module for the BMC REORG PLUS utility. 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.

**BMC_REORG_XBMID**

This keyword 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.

**BMC_UNLOAD_LOAD**

This keyword 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.
BMC_UNLOAD_OPTS=ADU$OPTS

This keyword 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.

CAT_LOAD

This keyword specifies the name of the LINK library for the BMC CATALOG MANAGER product.

CHECK+_LOAD

This keyword 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.

CHECKDOPT=ACK$MMS

This keyword 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.

CHGMAN_LOAD

This keyword specifies the name of the LINK library for CHANGE MANAGER.

TIP

To indicate the data set name for a different SSID, append the &SSID or &MSSID symbolic variable to the name.
**CLEANUP_RC=4**

This keyword specifies the value of the return code from the JCL cleanup job step. 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.

**CNTL_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the control data set.

**CNTL_EXPDT**

This keyword specifies the expiration date of the control data set on tape. A data set cannot have an expiration date and a retention period. The valid values are `yyddd` or `yyyy/ddd`.

**CNTL_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the control data set.

**CNTL_PREFIX=&PREFIX..&WKID..&SSID**

This keyword specifies the prefix for the name of the control data set that the BMC UNLOAD PLUS utility uses.

**CNTL_PRIQTY=1**

This keyword specifies the primary allocation (in cylinders) for the control data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**CNTL_RETPD**

This keyword specifies the retention period for the control data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

**CNTL_SECQTY=1**

This keyword specifies the secondary allocation (in cylinders) for the control data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 to 99999.
CNTL_STORCLASS

This keyword specifies the SMS definition for the storage class associated with the control data set.

CNTL_UNIT=SYSDA

This keyword specifies the unit name for the control data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

CNTLMOUT_DSN=&PREFIX..&SSID..CNTLMOUT(&JOBNAME)

For DASD MANAGER PLUS, this keyword specifies the name of the output data set that contains the job schedule name and the job sequence number of the non-IETFBR14 jobs.

CNTLMSCH_DSN=&PREFIX..&SSID..CNTLMSCH(&JOBNAME)

For DASD MANAGER PLUS, this keyword specifies the name of the input data set that contains the BMC Control-M job schedule.

COPY+_LOAD

This keyword specifies the name of the LINK library for the BMC COPY PLUS utility. 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.

COPYDOPT=ACP$MMS

This keyword specifies the name of the installation options module for the BMC COPY PLUS utility. 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.

CPYEXP_DATACLASS

This keyword specifies the SMS definition for the data class associated with the EXPORT data set that the BMC COPY PLUS EXPORT command creates to migrate data.

CPYEXP_EXPDT

This keyword specifies the expiration date of the EXPORT data set on tape that the BMC COPY PLUS EXPORT command creates to migrate data. A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.
**CPYEXP_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the EXPORT data set that the BMC COPY PLUS EXPORT command creates to migrate data.

**CPYEXP_PREFIX=&PREFIX..&WKID**

This keyword specifies the prefix for the name of the EXPORT data set that the BMC COPY PLUS EXPORT command creates to migrate data.

**CPYEXP_RETPD**

This keyword specifies the retention period for the EXPORT data set on tape that the BMC COPY PLUS EXPORT command creates to migrate data. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

**CPYEXP_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the EXPORT data set that the BMC COPY PLUS EXPORT command creates to migrate data.

**CPYEXP_SUPPRESS_SUFF=N**

This keyword specifies whether to suppress adding the DD name to the end of the name of the EXPORT data set that the BMC COPY PLUS EXPORT command creates to migrate data (Y or N). If you specify Y, you must ensure that the data set name is unique.

**CPYEXP_UNIT=SYSDA**

This keyword specifies the unit name for the EXPORT data set that the BMC COPY PLUS EXPORT command creates to migrate data. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**DASD_LOAD**

This keyword specifies the name of the LINK library for the BMC DASD MANAGER PLUS product.

--- **TIP**

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

This keyword specifies the name of the installation options module for the BMC DASD MANAGER PLUS product. ALTER and CHANGE MANAGER can use the BMCSTATS utility to collect statistics and populate the DASD MANAGER PLUS database, which is maintained in DB2 tables. If you select to use the BMCSTATS utility, you must specify a value for this keyword.

---

**DATA PACKER_LOAD**

This keyword specifies the name of the LINK library for the BMC DATA PACKER product.

---

**TIP**

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

---

**DATASETSIZING=N**

This keyword specifies the type of data set sizing. The following values are valid:

- **N**: does not perform data set sizing
- **C**: uses IBM RUNSTATS to perform data set sizing by using statistics from the DB2 catalog
- **B**: uses BMCSTATS to perform data set sizing by using the statistics from the BMC DASD MANAGER PLUS product tables
- **O**: physically and randomly samples the VSAM objects to estimate data set sizes

---

**DATAWK_NBR=4**

For CATALOG MANAGER, this keyword specifies the number of DATAWK data sets that the IBM REORG utility conditionally uses for sorting data.

---

**DATAWK_UNIT=SYSDA**

For CATALOG MANAGER, this keyword specifies the unit name of the DATAWK data set that the IBM REORG utility conditionally uses for sorting data. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

---

**DB2EXIT**

This keyword specifies the name of the DB2 EXIT library.
**DB2LOAD**

This keyword specifies the name of the DB2 LOAD library.

---

**DEF_GDG_BASE=N**

This keyword specifies whether to create the base of the generation data group (GDG) at JCL generation time (Y or N).

**DEF_GDG_LIMIT=10**

This keyword specifies the maximum number of GDG data sets that are allowed for primary copies. Valid values are 1 through 255.

**DEF_GDG_NOSCR=N**

This keyword specifies whether the base of a generation data group (GDG) is defined in the IDCAMS DEFINE command as EMPTY (NSCR). If the GDG is defined as EMPTY (NSCR), 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.

**DEF_GDG2_LIMIT=10**

This keyword specifies the maximum number of GDG data sets that are allowed for recovery copies. Valid values are 1 through 255.

**DIAG_MSGCLASS**

This keyword specifies the SYSOUT class that the components use for reporting incorrect entries in the POF. 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.

---

**TIP**

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

This keyword specifies the SMS definition for the discard data set’s data class.

**DISC_DATACLASS_ALT**

This keyword specifies the SMS definition for the discard data set’s data class (used if the threshold is exceeded).

**DISC_EXPDT**

This keyword specifies the expiration date of the discard data set on tape. A data set cannot have an expiration date and a retention period. The valid values are *yyddd* or *yyyy/ddd*.

**DISC_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the discard data set.

**DISC_MGMTCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the discard data set (used if the threshold is exceeded).

**DISC_PREFIX=&PREFIX..&OBNO</ref>**

This keyword specifies the prefix for the name of the discard data set.

**DISC_PRIQTY=10**

This keyword specifies the primary allocation (in cylinders) for the discard data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**DISC_RETPD**

This keyword specifies the retention period for the discard data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

**DISC_SECQTY=2**

This keyword 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.
**DISC_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the discard data set.

**DISC_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the discard data set (used if the threshold is exceeded).

**DISC_THRESH=0**

This keyword 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.

**DISC_UNIT=SYSDA**

This keyword specifies the unit for the discard data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**DISC_UNIT_ALT**

This keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**DISP_ALLOW_POPUP=N**

For ALTER and CHANGE MANAGER, this keyword 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
**DISP_AUTO_TAB=+**

For ALTER and CHANGE MANAGER, this keyword 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

**DISP_LOCATION=M**

For ALTER, CHANGE MANAGER, and DASD MANAGER PLUS, this keyword specifies the location of characters to be omitted in an object name that is too long to be displayed.

The following values are valid:

- B replaces characters at the left end (beginning) of the name
- M replaces characters in the middle of the name
- E replaces characters at the right end (end) of the name

**DISP_OMIT_CHAR=<>**

For ALTER, CHANGE MANAGER, and DASD MANAGER PLUS, this keyword specifies the characters that replace the beginning and end of a truncated string in an object name that is too long to be displayed.

**DISP_STATS=N**

This keyword 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 (Y or N).

**DISP_VAR_DBUG=N**

This keyword 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 (Y or N).

**DROPR_NOIC=N**

For ALTER, CATALOG MANAGER, and CHANGE MANAGER, this keyword specifies whether to invoke the Drop Recovery feature of the BMC CATALOG MANAGER product and drop an object (Y or N).

- N does not allow an object to be dropped, if no image copies of the object exist
- Y allows an object to be dropped, even if no image copies of the object exist
DSNCHECK44=N

This keyword specifies whether to verify that the prefix of a data set name contains 44 characters (Y or N). 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:

- You are modifying an SLIB because the ddnames that BMC generated do not meet your environment’s standards.
- You are creating the name of a data set.
- The data set name does not refer to a generation data group (GDG).

If you set the value to Y, you must ensure that the data set names are unique.

DSNTIAD_PLAN

This keyword specifies the name of the DB2 plan to run the IBM DSNTIAD program.

ERR_DATACLASS

This keyword specifies the SMS definition for the data class associated with the error data set.

ERR_DATACLASS_ALT

This keyword specifies the SMS definition for the data class associated with the error data set (used if the threshold is exceeded).

ERR_EXPDT

This keyword specifies the expiration date of the error data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.

ERR_MGMTCLASS

This keyword specifies the SMS definition for the storage class associated with the error data set.

ERR_MGMTCLASS_ALT

This keyword specifies the SMS definition for the storage class associated with the error data set (used if the threshold is exceeded).
**ERR_PREFIX=\&PREFIX..\&WKID..\&STEPN**

This keyword specifies the prefix for the name of the error data set.

**ERR_PRIQTY=10**

This keyword specifies the primary allocation (in cylinders) for the error data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**ERR_RETPD**

This keyword specifies the retention period for the error data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

**ERR_SECQTY=2**

This keyword 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.

**ERR_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the error data set.

**ERR_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the error data set (used if the threshold is exceeded).

**ERR_THRESH=0**

This keyword 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.

**ERR_UNIT=SYSDA**

This keyword specifies the unit for the error data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
ERR_UNIT_ALT

This keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

EXEC_LOAD

This keyword specifies the name of the Execution LINK library.

TIP

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

FCPY_DATACLASS

For CATALOG MANAGER, this keyword 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).

FCPY_EXPDT

For CATALOG MANAGER, this keyword specifies the expiration date of the flashcopy copy 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 yyddd or yyyy/ddd.

FCPY_MGMTCCLASS

For CATALOG MANAGER, this keyword 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).

FCPY_PREFIX=&PREFIX..&OBNO...P&PART

For CATALOG MANAGER, this keyword 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).

FCPY_PRIQTY=10

For CATALOG MANAGER, this keyword 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.
FCPY_RETPD

For CATALOG MANAGER, this keyword 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). A data set cannot have an expiration date and a retention period. Valid values are 1 to 9999.

FCPY_SECQTY=2

For CATALOG MANAGER, this keyword specifies the secondary allocation (in cylinders) for the flashcopy data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume).

FCPY_STORCLASS

For CATALOG MANAGER, this keyword 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).

FCPY.Suppress_Suff=N

For CATALOG MANAGER, this keyword specifies whether to suppress adding the DD name to the end of the name of 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).

FCPY_UNIT=SYSDA

For CATALOG MANAGER, this keyword specifies the unit for the flashcopy data set. The IBM FlashCopy feature creates the flashcopy (a point-in-time copy of a volume). The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

FILT_DATACLASS

This keyword specifies the SMS definition for the data class associated with the filter data set.

FILT_EXPDT

This keyword 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 yyddd or yyyy/ddd.
**Product options**

**FILT_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the filter data set.

**FILT_PREFIX=&PREFIX..&WKID..&STEPN**

This keyword specifies the prefix for the name of the filter data set.

**FILT_PRIQTY=10**

This keyword specifies the primary allocation (in cylinders) for the filter data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**FILT_RETPD**

This keyword specifies the retention period for the filter data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

**FILT_SECQTY=2**

This keyword specifies the secondary allocation (in cylinders) for the filter data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 9999.

**FILT_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the filter data set.

**FILT_UNIT=SYSDA**

This keyword specifies the unit name for the filter data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**GDG_MODEL=SYS1.MODEL**

This keyword specifies the name of the GDG model data set.

**HASHFAIL=N**

This keyword specifies whether Execution terminates a job if a hash failure, such as a changed or added statement, occurs in a worklist.
**HASWARNRC**

This keyword defines the return code that the product sends back when the product finds only hash warnings.

---

**NOTE**

Do not use 8 for this value.

---

**INCLUDE_SYSPRIN2=N**

This keyword specifies whether to add the following DD to execution JCL for utility jobs:

```
//SYSPRIN2 DD SYSOUT=*  
```

The SYSPRIN2 output data set contains SYSPRINT output messages for versions 10.2 and later of the following BMC utilities:

- CHECK PLUS
- LOADPLUS
- REORG PLUS
- UNLOAD PLUS

If you select Y, you can view the SYSPRINT output from a utility while an execution job runs the utility or when an execution job cancels during the running of the utility.

---

**NOTE**

SYSPRIN2 data sets have the following restrictions:

- When you specify BMCSTATS YES or UPDATEDB2STATS YES for LOADPLUS or REORG PLUS, SYSPRIN2 does not contain the statistics report from the Common Statistics component.

- When invoking the IBM DSNUTILB utility, REORG PLUS and LOADPLUS ignore the SYSPRIN2 DD statement.

---

**IOALOAD1**

This keyword specifies the name of a LINK library for the utility automation component of the BMC Database Performance for DB2 solution.
**Product options**

---

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

---

**IOALOAD2**

This keyword specifies the name of a LINK library for the utility automation component of the BMC Database Performance for DB2 solution.

---

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

---

**JCLCLEANUP=N**

This keyword specifies whether to generate a job step that automatically deletes many of the permanent work data sets that Execution creates (Y or N).

---

**JCLLIB**

This keyword specifies the name of a partitioned data set (PDS) that contains JCL to be included in a job, or the name of a PDS that specifies the cataloged procedures (PROCs) that are used for non-worklist JCL.

---

**JES3=N**

JCL Generation no longer uses this keyword.

---

**JOB_INCLUDE_MEMBER**

This keyword specifies the name of a JCL member to be included at the end of a job.

```plaintext
JOBCARD1=>
//JOBC JOB (&ZACCTNUM),’&PGMR’,
JOBCARD2=// CLASS=A,MSGLEVEL=(1,1)
JOBCARD3=//*
JOBCARD4=//*
JOBCARD5=//*
```

These keywords define the default job card statement that the components use when JCL Generation generates Analysis and Execution JCL. For DASD MANAGER PLUS, these keywords specify the BMCTRIG and batch report job cards.
LISTDEF_DSN

For CATALOG MANAGER, this keyword specifies the name of the data set that contains member names for LISTDEF utility control statements.

LL_CLIB=CLIB
LL_CLIB2
LL_CLIB3
LL_CLIB4
LL_CLIB5

These keywords specify the low-level qualifier (LLQ) for the product CLIST data sets for the batch ISPF environment.

LL_LINK=LINK
LL_LINK2
LL_LINK3
LL_LINK4
LL_LINK5

These keywords specify the LLQ for the LOAD library data sets for the batch ISPF environment.

LL_MLIB=MLIB
LL_MLIB2
LL_MLIB3
LL_MLIB4
LL_MLIB5

These keywords specify the LLQ for the message data sets for the batch ISPF environment.

LL_PLIB=PLIB
LL_PLIB2
LL_PLIB3
LL_PLIB4
LL_PLIB5

These keywords specify the LLQ for the panel and Help library data sets for the batch ISPF environment.
**Product options**

**LL_SLIB=SLIB**
**LL_SLIB2**
**LL_SLIB3**
**LL_SLIB4**
**LL_SLIB5**

These keywords specify the LLQ for the ISPF skeleton data sets for the batch ISPF environment.

**LL_TLIB=TLIB**
**LL_TLIB2**
**LL_TLIB3**
**LL_TLIB4**
**LL_TLIB5**

These keywords specify the LLQ for the ISPF table data sets for the batch ISPF environment.

**LL_XML=XML**
**LL_XML2**
**LL_XML3**
**LL_XML4**
**LL_XML5**

These keywords specify the LLQ for the utility generation data sets for the batch ISPF environment.

**LLQ**

This keyword specifies the LLQ for ISPF data sets for the batch ISPF 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.

---

**NOTE**

Changing the qualifier of the ISPF data sets might cause unpredictable results. Do not change the LLQ for the ISPF data sets.

---

**LOAD+_LOAD**

This keyword specifies the name of the LINK library for the BMC LOADPLUS utility. 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.
LOADDOPT

This keyword specifies the name of the installation options module for the BMC LOADPLUS utility. 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.

LOGWK_NBR=4

This keyword specifies the number of LOGSORT data sets. Valid values are 1 through 32.

LOGWK_UNIT=SYSDA

This keyword specifies the unit name of the LOGSORT data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

MAP_DATACLASS

This keyword specifies the SMS definition for the data class associated with the map data set.

MAP_DATACLASS_ALT

This keyword specifies the SMS definition for the data class associated with the map data set (used if the threshold is exceeded).

MAP_EXPDT

This keyword specifies the expiration date of the map data set on tape. A data set cannot have an expiration date and a retention period. The valid values are yyddd or yyyy/ddd.

MAP_MGMTCLASS

This keyword specifies the SMS definition for the storage class associated with the map data set.

MAP_MGMTCLASS_ALT

This keyword specifies the SMS definition for the storage class associated with the map data set (used if the threshold is exceeded).

MAP_PREFIX=&PREFIX..&WKID..&SSID

This keyword specifies the prefix for the name of the map data set.
**MAP_PRIQTY=10**

This keyword 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.

**MAP_RETPD**

This keyword 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.

**MAP_SECQTY=2**

This keyword 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.

**MAP_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the map data set.

**MAP_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the map data set (used if the threshold is exceeded).

**MAP_THRESH=0**

This keyword 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.

**MAP_UNIT=SYSDA**

This keyword specifies the unit for the map data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**MAP_UNIT_ALT**

This keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
**MAX_CYL=99999**

This keyword 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.

**MAX_PRIQTY=2000**

This keyword specifies the primary quantity in cylinders that will be used when the value of MAX_CYL is reached. Valid values are 1 through 9999.

**MAX_SECQTY=200**

This keyword specifies the secondary quantity in cylinders that will be used when the value of MAX_CYL is reached. Valid values are 1 through 9999.

**MAX_UNITCNT**

This keyword specifies the value for the DASD unit count. Valid values are 1 through 59.

**MEMLIMIT**

This keyword specifies the limit on above-the-bar memory for an address space.

**ORTPARM_DSN**

This keyword specifies the name of the data set for the SyncSort parameters.

**PCPY1_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the local primary copy data set.

**PCPY1_DATACLASS_ALT**

This keyword specifies the SMS definition for the data class associated with the local primary copy data set (used if the threshold is exceeded).

**PCPY1_EXPDT**

This keyword 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 `yyddd` or `yyyy/ddd`.
**PCPY1_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the local primary copy data set.

**PCPY1_MGMTCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the local primary copy data set (used if the threshold is exceeded).

**PCPY1_PREFIX=&PREFIX..&OBNOD..P&PART**

This keyword specifies the prefix for the name of the local primary copy data set.

**PCPY1_PRIQTY=10**

This keyword specifies the primary allocation (in cylinders) for the local primary copy data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**PCPY1_RETPD**

This keyword 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.

**PCPY1_SECQTY=2**

This keyword specifies the secondary allocation (in cylinders) for the local primary copy data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**PCPY1_STACK=N**

This keyword specifies whether to stack the local primary copy data set on a tape with data sets of the same type (Y or N).

**PCPY1_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the local primary copy data set.

**PCPY1_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the local primary copy data set (used if the threshold is exceeded).
PCPY1_SUPPRESS_SUFF=N

This keyword specifies whether to suppress adding the DD name to the end of the name of the local primary copy data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

PCPY1_THRESH=0

This keyword 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.

PCPY1_UNIT=SYSDA

This keyword specifies the unit for the local primary copy data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

PCPY1_UNIT_ALT

This keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

PCPY2_DATACLASS

This keyword specifies the SMS definition for the data class associated with the local backup copy data set.

PCPY2_DATACLASS_ALT

This keyword specifies the SMS definition for the data class associated with the local backup copy data set (used if the threshold is exceeded).

PCPY2_EXPDT

This keyword 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.

PCPY2_MGMTCLASS

This keyword specifies the SMS definition for the storage class associated with the local backup copy data set.
**PCPY2_MGMTCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the local backup copy data set (used if the threshold is exceeded).

**PCPY2_PREFIX=&PREFIX..&OBNOQ..P&PART**

This keyword specifies the prefix for the name of the local backup copy data set.

**PCPY2_PRIQTY=10**

This keyword 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.

**PCPY2_RETPD**

This keyword 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.

**PCPY2_SECQTY=2**

This keyword 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 99999.

**PCPY2_STACK=N**

This keyword specifies whether to stack the local backup copy data set on a tape with data sets of the same type (Y or N).

**PCPY2_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the local backup copy data set.

**PCPY2_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the local backup copy data set (used if the threshold is exceeded).

**PCPY2_SUPPRESS_SUFF=N**

This keyword specifies whether to suppress adding the DD name to the end of the name of the local backup copy data set (Y or N). If you specify Y, you must ensure that the data set name is unique.
PCPY2_THRESH=0

This keyword 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.

PCPY2_UNIT=SYSDA

This keyword specifies the unit for the local backup copy data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

PCPY2_UNIT_ALT

This keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

POFDATE

This keyword shows the last date on which the initial product options file (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.

PRE_JOBSTEP_INCLUDE

This keyword specifies the name of a JCL member to be included before each step in the JCL.

PROC_BMCCHECK_NAME

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.

PROC_BMCCHECK_STEP

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.
**PROC_BMCCOPY_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.

**PROC_BMCCOPY_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.

**PROC_BMCCPRS_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.

**PROC_BMCCPRS_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.

**PROC_BMCLOAD_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.

**PROC_BMCLOAD_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC LOADPLUS for DB2 utility.
This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCRECOVER_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC RECOVER PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCRECOVER_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC RECOVER PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCREORG_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC REORG PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCREORG_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC REORG PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCSTATS_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCSTATS_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.
This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCSTOP_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility. BMCSTOP refers to the part of the utility that issues a DB2 STOP command on an object and verifies the completion of the command.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCSTOP_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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. BMCSTOP refers to the part of the utility that issues a DB2 STOP command on an object and verifies the completion of the command.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCTRIG_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCTRIG_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCUNLOAD_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC UNLOAD PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.
**PROC_BMCUNLOAD_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the BMC UNLOAD PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCUPRS_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the BMC DASD MANAGER PLUS for DB2 utility. BMCUPRS refers to the part of 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.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCUPRS_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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. BMCUPRS refers to the part of 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.

This keyword is not included in the AJXPOFIN input stream.

**PROC_DSNUTILB_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM DSNUTILB utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_DSNUTILB_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM DSNUTILB utility.

This keyword is not included in the AJXPOFIN input stream.
**PROC_DSN1COPY_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM DSN1COPY utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_DSN1COPY_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM DSN1COPY utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_GEN_SET_VAR=N**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies whether to include an SLIB that generates SET statements in the JCL for variables that you can use in catalog procedures (PROCs).

This keyword is not included in the AJXPOFIN input stream.

**PROC_IDCAMS_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM IDCAMS program.

This keyword is not included in the AJXPOFIN input stream.

**PROC_IDCAMS_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM IDCAMS program.

This keyword is not included in the AJXPOFIN input stream.

**PROC_IEFBR14_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for the IBM IEFBR14 job.

This keyword is not included in the AJXPOFIN input stream.
**PROC_IEFBR14_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for the IBM IEFBR14 job.

This keyword is not included in the AJXPOFIN input stream.

**PROC_TSO_NAME**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the cataloged procedure (PROC statement) for non-worklist JCL for IBM TSO.

This keyword is not included in the AJXPOFIN input stream.

**PROC_TSO_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword specifies the name of the EXEC job step in the cataloged procedure for non-worklist JCL for IBM TSO.

This keyword is not included in the AJXPOFIN input stream.

**PROC_USE=N**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.

**PROC_USER_DEF_STEP**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.

**PROC_USER_DEFINED**

For CATALOG MANAGER and DASD MANAGER PLUS, this keyword 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.
PUNCH_DATACLASS

This keyword specifies the SMS definition for the data class associated with the punch data set.

PUNCH_EXPDT

This keyword 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 $yyyy/ddd$.

PUNCH_MGMTCLASS

This keyword specifies the SMS definition for the storage class associated with the punch data set.

PUNCH_PREFIX=$PREFIX..&WKID..&STEPN

When a limit key is changed, this keyword 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.

PUNCH_PRIQTY=1

This keyword 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.

PUNCH_RETPD

This keyword 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.

PUNCH_SECQTY=1

This keyword 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.

PUNCH_STORCLASS

This keyword specifies the SMS definition for the storage class associated with the punch data set.
PUNCH_UNIT=SYSDA

This keyword specifies the unit name for the punch data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

RCPY1_DATACLASS

This keyword specifies the SMS definition for the data class associated with the recovery primary copy data set.

RCPY1_DATACLASS_ALT

This keyword specifies the SMS definition for the data class associated with the recovery primary copy data set (used if the threshold is exceeded).

RCPY1_EXPDT

This keyword 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.

RCPY1_MGMTCLASS

This keyword specifies the SMS definition for the storage class associated with the recovery primary copy data set.

RCPY1_MGMTCLASS_ALT

This keyword specifies the SMS definition for the storage class associated with the recovery primary copy data set (used if the threshold is exceeded).

RCPY1_PREFIX=&PREFIX..&OBNOD..P&PART

This keyword specifies the prefix for the name of the recovery primary copy data set.

RCPY1_PRIQTY=10

This keyword 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.

RCPY1_RETPD

This keyword 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.
**RCPY1_SECQTY=2**

This keyword 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.

**RCPY1_STACK=N**

This keyword specifies whether to stack the recovery primary copy data set on a tape with data sets of the same type (Y or N).

**RCPY1_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the recovery primary copy data set.

**RCPY1_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the recovery primary copy data set (used if the threshold is exceeded).

**RCPY1_SUPPRESS_SUFF=N**

This keyword specifies whether to suppress adding the DD name to the end of the name of the recovery primary copy data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

**RCPY1_THRESH=0**

This keyword 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.

**RCPY1_UNIT=SYSDA**

This keyword specifies the unit for the recovery primary copy data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**RCPY1_UNIT_ALT**

This keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
**RCPY2_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the recovery backup copy data set.

**RCPY2_DATACLASS_ALT**

This keyword specifies the SMS definition for the data class associated with the recovery backup copy data set (used if the threshold is exceeded).

**RCPY2_EXPDT**

This keyword 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 *yyddd* or *yyyy/ddd*.

**RCPY2_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the recovery backup copy data set.

**RCPY2_MGMTCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the recovery backup copy data set (used if the threshold is exceeded).

**RCPY2_PREFIX=&PREFIX..&OBNOD..P&PART**

This keyword specifies the prefix for the name of the recovery backup copy data set.

**RCPY2_PRIQTY=10**

This keyword 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.

**RCPY2_RETPD**

This keyword 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.

**RCPY2_SECQTY=2**

This keyword 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.
**RCPY2_STACK=N**

This keyword specifies whether to stack the recovery backup copy data set on a tape with data sets of the same type (Y or N).

**RCPY2_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the recovery backup copy data set.

**RCPY2_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the recovery backup copy data set (used if the threshold is exceeded).

**RCPY2_SUPPRESS_SUFF=N**

This keyword specifies whether to suppress adding the DD name to the end of the name of the recovery backup copy data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

**RCPY2_THRESH=0**

This keyword 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.

**RCPY2_UNIT=SYSDA**

This keyword specifies the unit for the recovery backup copy data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**RCPY2_UNIT_ALT**

This keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
### REBINDFAIL=N

This keyword 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.

### REBINDRC

This keyword allows worklist execution to continue if a rebind fails, but returns a value for a final condition code instead of 4, the default value. Execution writes warning messages to AEXPRINT but does not post entries in the sync tables.

### RECOVER+_LOAD

This keyword specifies the name of the LINK library for the BMC RECOVER PLUS utility. 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`.

### RECOVERDOPT

This keyword specifies the name of the installation options module for the BMC RECOVER PLUS utility. 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`.

### REGION=0M

This keyword defines the REGION parameter in the EXEC statement.

### REORG_MAPTAB

This keyword specifies the name of the mapping table that the IBM REORG or BMC REORG PLUS utility uses to map the row IDs (RIDs) in the source table to the RIDs in the target table. 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, and can contain the &ZUSER or &USERID symbolic variable.

### NOTE

The REORG PLUS utility invokes the IBM DSNUTILB utility control program to enable certain features. If you have specified to use the REORG PLUS utility, you still need to specify mapping table information. For information about the features for which REORG PLUS invokes DSNUTILB, see the REORG PLUS for DB2 Reference Manual.
**REORG+_LOAD**

This keyword specifies the name of the LINK library for the BMC REORG PLUS utility. 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.

**REORGDOPT**

This keyword specifies the name of the installation options module for the BMC REORG PLUS utility. 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.

**REPT_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the report data set.

**REPT_DATACLASS_ALT**

This keyword specifies the SMS definition for the data class associated with the report data set (used if the threshold is exceeded).

**REPT_EXPDT**

This keyword specifies the expiration date of the report data set on tape. A data set cannot have an expiration date and a retention period. The valid values are `yyddd` or `yyyy/ddd`.

**REPT_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the report data set.

**REPT_MGMTCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the report data set (used if the threshold is exceeded).

**REPT_PREFIX=&PREFIX..&WKID**

This keyword specifies the prefix for the name of the report data set.
**REPT_PRIQTY=10**

This keyword specifies the primary allocation (in cylinders) for the report data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**REPT_RETPD**

This keyword specifies the retention period for the report data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

**REPT_SECQTY=2**

This keyword specifies the secondary allocation (in cylinders) for the report data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**REPT_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the report data set.

**REPT_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the report data set (used if the threshold is exceeded).

**REPT_THRESH=0**

This keyword specifies the maximum anticipated size for the report 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.

**REPT_UNIT=SYSDA**

This keyword specifies the unit for the report data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**REPT_UNIT_ALT**

This keyword specifies the alternate unit name for the report data set, if the threshold value is exceeded. The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
RUNTIME_HLQ

This keyword specifies a high-level qualifier (HLQ) for ISPF data sets for the batch
ISPF environment. 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.

SCHED_TRIG_CTLM_JOBS=N

This keyword specifies whether JCL Generation should generate BMC Control-M job
schedule entries for jobs that do not contain IEFBR14 steps (Y or N). If the value is Y,
JCL Generation generates an input data set (CNTLMSCH) and an output data set
(CNTLMOUT) in the JCL for the BMCTRIG utility.

SORTWK_NBR=4

This keyword specifies the number of SORTWORK data sets. Valid values are 1
through 32.

SORTWK_PRIQTY=10

This keyword specifies the primary allocation (in cylinders) for the DATAWORK,
LOGSORT, or SORTWORK data set if DATASETSIZING=N or if an error in sizing
occurs. Valid values are 1 through 99999.

SORTWK_SECQTY=2

This keyword specifies the secondary quantity (in cylinders) for the DATAWORK,
LOGSORT, or SORTWORK data set if DATASETSIZING=N or if an error in sizing
occurs. Valid values are 1 through 99999.

SORTWK_UNIT=SYSDA

This keyword specifies the name of the SORTWORK data set. The value of the unit
can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT
parameter from the JCL, specify NONE.

SQLEXP_LOAD

This keyword specifies the name of the LINK library for the BMC SQL Explorer for
DB2 product.

TIP

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

This keyword specifies the SMS definition for the data class associated with the SORTOUT data set.

**SRTOUT_DATACLASS_ALT=CART**

This keyword specifies the SMS definition for the data class associated with the SORTOUT data set (used if the threshold is exceeded).

**SRTOUT_EXPDT**

This keyword specifies the expiration date of the SORTOUT data set on tape. A data set cannot have an expiration date and a retention period. The valid values are `yyddd` or `yyyy/ddd`.

**SRTOUT_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the SORTOUT data set.

**SRTOUT_MGMTCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the SORTOUT data set (used if the threshold is exceeded).

**SRTOUT_PREFIX=&PREFIX..&WKID..&STEPN**

This keyword specifies the prefix for the name of the SORTOUT data set.

**SRTOUT_PRIQTY=10**

This keyword specifies the primary allocation (in cylinders) for the SORTOUT data set if `DATASETSIZING=N` or if an error in sizing occurs. Valid values are 1 through 99999.

**SRTOUT_RETPD=**

This keyword specifies the retention period for the SORTOUT data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

**SRTOUT_SECQTY=2**

This keyword specifies the secondary allocation (in cylinders) for the SORTOUT data set if `DATASETSIZING=N` or if an error in sizing occurs. Valid values are 1 through 99999.
**SRTOUT_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the SORTOUT data set.

**SRTOUT_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the SORTOUT data set (used if the threshold is exceeded).

**SRTOUT_THRESH=0**

This keyword specifies the maximum anticipated size for the SORTOUT 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.

**SRTOUT_UNIT=SYSDA**

This keyword specifies the unit for the SORTOUT data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**SRTOUT_UNIT_ALT=CART**

This keyword specifies the alternate unit name for the SORTOUT data set, if the threshold value is exceeded. The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**STEP_INCLUDE_MEMBER**

This keyword specifies the name of a JCL member to be included after each step in the JCL.

**STOPWAIT=3**

This keyword 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).

**STOPWTSECS=10**

This keyword 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.
**SUPPRESS_COMMENTS=N**

This keyword specifies whether to suppress the comments in the generated JCL (Y or N).

**SYNCDELETE=N**

This keyword specifies whether Execution should remove all sync entries when an Execution job completes with no errors (Y or N).

**SYSEXEC**

This keyword specifies the name of the partitioned data set 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.

**SYSTEM_MLIB**

This keyword specifies the name of the system ISPF message library. You specify the value for this keyword during installation.

**SYSUT_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the SYSUT data set.

**SYSUT_DATACLASS_ALT**

This keyword specifies the SMS definition for the data class associated with the SYSUT data set (used if the threshold is exceeded).

**SYSUT_EXPDT**

This keyword 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.

**SYSUT_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the SYSUT data set.
SYSUT_MGMTCLASS_ALT

This keyword specifies the SMS definition for the storage class associated with the SYSUT data set (used if the threshold is exceeded).

SYSUT_PREFIX=&PREFIX..&WKID..&STEPN

This keyword specifies the prefix for the name of the SYSUT and WORKDDN data sets.

SYSUT_PRIQTY=10

This keyword 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.

SYSUT_RETPD

This keyword 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.

SYSUT_SECQTY=2

This keyword 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.

SYSUT_STORCLASS

This keyword specifies the SMS definition for the storage class associated with the SYSUT data set.

SYSUT_STORCLASS_ALT

This keyword specifies the SMS definition for the storage class associated with the SYSUT data set (used if the threshold is exceeded).

SYSUT_THRESH=0

This keyword 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.
**SYSUT_UNIT=SYSDA**

This keyword specifies the unit for the SYSUT data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**SYSUT_UNIT_ALT**

This keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**SZDEVT=3390**

This keyword specifies the device type used in data set sizing. Valid values are 3380 or 3390.

**TAPE_EXPDT**

This keyword 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.

**TAPE_RETPD**

This keyword 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.

**TAPE_VOLCNT**

This keyword specifies the maximum number of tape volumes. Valid values are 0 through 255.

**TAPE1=CART**
**TAPE2=TAPE**
**TAPE3=TAPE**

These keywords define the names of the tape units for an installation.

**TEMPLATE_DSN**

For CATALOG MANAGER, this keyword specifies the name of the data set that contains member names for TEMPLATE utility control statements.
**Product options**

### TEMPUNIT=SYSDA

This keyword defines the name of the unit that the components use for temporary files. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

### TIMEPARAM

This keyword indicates the time limit in minutes for each step in a batch job stream.

### TRTCH

This keyword specifies the parity, data conversion, translation, and compression for 7-track drives. The following values are valid:

- **B**: 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

### TSOPROGRAM

This keyword specifies an alternate TSO monitor program for standard JCL. TSOPROGRAM is available for nonworklist JCL.

### TSOSUBEXIT=N

This keyword specifies whether to use a TSO submit exit to generate job cards. If **TSOSUBEXIT=Y**, no job cards are put in the JCL (Y or N).

### ULLQ

This keyword specifies the LLQ 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, this keyword 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, this keyword specifies the number of directory blocks for the file reference (SYSREC) data set.

**UNLD_FREF_MGMTCLASS**

For ALTER and CHANGE MANAGER, this keyword specifies the SMS definition for the storage class associated with the file reference (SYSREC) data set.

**UNLD_FREF_PREFIX = &PREFIX..&MSSID..&WORKID8**

For ALTER and CHANGE MANAGER, this keyword specifies the prefix for the name of the file reference (SYSREC) data set.

**UNLD_FREF_PRIQTY = 10**

For ALTER and CHANGE MANAGER, this keyword 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, this keyword 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, this keyword specifies the SMS definition for the storage class associated with the file reference (SYSREC) data set.

**UNLD_FREF_SUPPR_SUFF=N**

For ALTER and CHANGE MANAGER, this keyword specifies whether to suppress adding the DD name to the end of the name of the file reference (SYSREC) data set (Y or N). If you specify Y, you must ensure that the data set name is unique.
**UNLD_FREF_UNIT = SYSDA**

For ALTER and CHANGE MANAGER, this keyword specifies the unit for the file reference (SYSREC) data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**UNLD1_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the primary unload (SYSREC) data set.

**UNLD1_DATACLASS_ALT**

This keyword specifies the SMS definition for the data class associated with the primary unload (SYSREC) data set (used if the threshold is exceeded).

**UNLD1_EXPDT**

This keyword 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 \( yyddd \) or \( yyyy/ddd \).

**UNLD1_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set.

**UNLD1_MGMTCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set (used if the threshold is exceeded).

**UNLD1_PREFIX=&USERID..&MSSID..&WORKID8**

This keyword specifies the prefix for the name of the primary unload (SYSREC) data set.

**UNLD1_PRIQTY=10**

This keyword 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.
**UNLD1_RETPD**

This keyword 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.

**UNLD1_SECQTY=2**

This keyword 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.

**UNLD1_STACK=N**

This keyword specifies whether to stack the primary unload (SYSREC) data set on a tape with data sets of the same type (Y or N).

**UNLD1_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set.

**UNLD1_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the primary unload (SYSREC) data set (used if the threshold is exceeded).

**UNLD1_SUPPRESS_SUFF=N**

This keyword specifies whether to suppress adding the DD name to the end of the name of the primary unload (SYSREC) data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

**UNLD1_THRESH=0**

This keyword 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.

**UNLD1_UNIT=SYSDA**

This keyword specifies the unit for the primary unload (SYSREC) data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
**UNLD1_UNIT_ALT**

This keyword specifies the alternate unit name for the primary unload (SYSREC) data set (used if the threshold value is exceeded). The unit name must be defined in the TAPE1, TAPE2, or TAPE3 keywords. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**UNLD2_DATACLASS**

This keyword specifies the SMS definition for the data class associated with the backup unload (SYSREC) data set.

**UNLD2_DATACLASS_ALT**

This keyword specifies the SMS definition for the data class associated with the backup unload (SYSREC) data set (used if the threshold is exceeded).

**UNLD2_EXPDT**

This keyword specifies the expiration date of the backup unload (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`.

**UNLD2_MGMTCLASS**

This keyword specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set.

**UNLD2_MGMTCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set (used if the threshold is exceeded).

**UNLD2_PREFIX=&USERID..&MSSID..&WORKID8**

This keyword specifies the prefix for the name of the backup unload (SYSREC) data set.

**UNLD2_PRIQTY=10**

This keyword specifies the primary allocation (in cylinders) for the backup unload (SYSREC) data set if `DATASETSIZING=N` or if an error in sizing occurs. Valid values are 1 through 99999.
**UNLD2_RETPD**

This keyword specifies the retention period for the backup unload (SYSREC) data set on tape. A data set cannot have an expiration date and a retention period. Valid values are 1 through 9999.

**UNLD2_SECQTY=2**

This keyword specifies the secondary allocation (in cylinders) for the backup unload (SYSREC) data set if DATASETSIZING=N or if an error in sizing occurs. Valid values are 1 through 99999.

**UNLD2_STACK=N**

This keyword specifies whether to stack the backup unload (SYSREC) data set on a tape with data sets of the same type (Y or N).

**UNLD2_STORCLASS**

This keyword specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set.

**UNLD2_STORCLASS_ALT**

This keyword specifies the SMS definition for the storage class associated with the backup unload (SYSREC) data set (used if the threshold is exceeded).

**UNLD2_SUPPRESS_SUFF=N**

This keyword specifies whether to suppress adding the DD name to the end of the name of the backup unload (SYSREC) data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

**UNLD2_THRESH=0**

This keyword 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.

**UNLD2_UNIT=SYSDA**

This keyword specifies the unit for the backup unload (SYSREC) data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.
**UNLD2_UNIT_ALT**

This keyword 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. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**UNLD3_DATACLASS**

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the SMS definition for the data class associated with the ROWID SYSREC data set.

**UNLD3_EXPDT**

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword 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.

**UNLD3_MGMTCLASS**

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the SMS definition for the storage class associated with the ROWID SYSREC data set.

**UNLD3_PREFIX=&USERID..&MSSID..&WORKID8**

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the prefix for the name of the ROWID SYSREC data set.

**UNLD3_RETPD**

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword 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.

**UNLD3_STORCLASS**

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the SMS definition for the storage class associated with the ROWID SYSREC data set.
**UNLD3_SUPPRESS_SUFF=N**

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies whether to suppress adding the DD name to the end of the name of the ROWID SYSREC data set (Y or N). If you specify Y, you must ensure that the data set name is unique.

**UNLD3_UNIT=SYSDA**

For the BMC UNLOAD PLUS utility in the Database Administration solution, this keyword specifies the unit for the ROWID SYSREC data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

**UNLD4_DATACLASS**

For the LOB DATA MOVER program in the Database Administration solution, this keyword specifies the SMS definition for the data class associated with the large object (LOB) SYSREC data set.

**UNLD4_MGMTCLASS**

For the LOB DATA MOVER program in the Database Administration solution, this keyword specifies the SMS definition for the storage class associated with the large object (LOB) SYSREC data set.

**UNLD4_PREFIX=&USERID..&MSSID..&WORKID8**

For the LOB DATA MOVER program in the Database Administration solution, this keyword specifies the prefix for the name of the large object (LOB) SYSREC data set.

**UNLD4_STORCLASS**

For the LOB DATA MOVER program in the Database Administration solution, this keyword specifies the SMS definition for the storage class associated with the large object (LOB) SYSREC data set.

**UNLD4_SUPPRESS_SUFF=N**

For the LOB DATA MOVER program in the Database Administration solution, this keyword specifies whether to suppress adding the DD name to the end of the name of the large object (LOB) SYSREC data set (Y or N). If you specify Y, you must ensure that the data set name is unique.
UNLD4_UNIT=SYSDA

For the LOB DATA MOVER program in the Database Administration solution, this keyword specifies the unit for the large object (LOB) SYSREC data set. The value of the unit can be a name from 1 to 8 characters long, blank, or NONE. To omit the UNIT parameter from the JCL, specify NONE.

UNLOAD+_LOAD

This keyword specifies the name of the LINK library for the BMC UNLOAD PLUS utility. 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.

UNLOADDOPT

This keyword specifies the name of the installation options module for the BMC UNLOAD PLUS utility. 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.

USER_VAR1_CHAR
USER_VAR2_CHAR
USER_VAR3_CHAR
USER_VAR4_CHAR
USER_VAR5_CHAR

These keywords specify user-defined character variables. Each variable has a corresponding symbolic variable that you can use in job cards or data set prefixes. The maximum length of a variable name is eight characters.

WORK_DATACLASS

This keyword defines the Data Facility Storage Management Subsystem (DFSMS or SMS) data class name that the components use at data-set allocation time to define allocation attributes of the data set. A data class name is not required, even for SMS data sets. This option appears as "DATACLAS=" in the JCL for workfiles.

WORK_MGMTCLASS

This keyword defines the DFSMS or SMS management class name that the components use at data-set allocation time to define the migration, retention, and backup requirements of the data set. This option appears as "MGMTCLAS=" in the JCL for workfiles.
WORK\_STORCLASS

This keyword defines the DFSMS or SMS storage class name that the components use at data-set allocation time to define processing requirements of the data set. This option appears as "STORCLAS=\" in the JCL for nontape work files.
CATALOG MANAGER worklist commands

This appendix presents the following topics:

Worklist overview ................................................................. 523
Worklist file format ............................................................... 524
Worklist commands ............................................................ 525
  -BMCU—Run a BMC utility ................................................. 525
  -DEBUG—Debug ................................................................. 526
  -DSN1—DSN1COPY (IBM utility) ......................................... 526
  -MERG—MERGECOPY (IBM utility) .................................... 527
  -MODI—MODIFY (IBM utility) ............................................. 528
  -NOOP—No operation ....................................................... 528
  -QUI—QUIESCE (IBM utility) ............................................. 529
  -REPO—REPORT (IBM utility) ............................................ 530
  -REPX—REPAIR (IBM utility) ............................................ 530
  -SQLM (Group multiple ALTER statements) ......................... 531
  -STOS—STOSPACE (IBM utility) ......................................... 531

Worklist overview

A worklist consists of a number of commands that the Execution component uses to set up and run DB2 SQL statements, DB2 commands and utilities, and BMC programs. Some of these commands are not generated by CATALOG MANAGER, but all of the commands are listed here for complete information. If you need to use one of these commands, you can manually add it to a worklist.
Worklist file format

Worklist files consist of a sequence of commands that occupy single or multiple lines. The file structure is sequential and fixed in length, with 80-character records. The following list defines the record layout of the worklist file:

1—Line identifier

A dash (-) indicates the beginning of a command. An asterisk (*) indicates a comment. If there is no indicator, then this line is a continuation of the previous Command line.

2-5—Command identifier

A four-character code that identifies the command. The command identifier appears only on the first line of the command.

6—Blank

This character must be blank.

7-12—Command sequence number

A six-digit number, right-justified and zero-filled indicates the command sequence. Each command has a unique number. These numbers must remain in order. You might add and delete commands, however, you must keep the commands in ascending sequence. Command sequence numbers are generated in increments of 50 to allow you to insert commands as needed. Although you can change a command sequence number, doing so results in a hash failure. The command sequence number appears only on the first line of a command.

13—Blank

This character must be blank.

14-72—Command text

Command text is free-format text that includes the information that is needed to execute the command. For continuation lines, the text appears in columns 2 through 72.

NOTE

No implicit break or space exists between column 72 of one line and column 2 of the following continuation line. Keywords cannot be split over multiple lines.
73-80—Hash verification number

A numerical hash value indicates the contents of the command. Several components use the number to detect changed and inserted commands. Do not insert or modify this code. The hash number appears only on the last line of a command.

Multi-line commands

Multi-line commands consist of the following identifiers and numbers:

- the line identifier, command identifier, command sequence number, and short command text on the first line
- a blank in the line identifier column, with long command text on continuation lines
- the hash verification number on the last line

Comment lines

Comment lines consist of an asterisk (*) for the line identifier and comment text for the remainder of the line. Comments do not use sequence numbers or hash verification numbers.

NOTE

You cannot embed comment lines within a command. A comment signals the end of the previous command.

Worklist commands

This section describes each command in detail and provides an example of the command.

-BMCU—Run a BMC utility

The -BMCU command invokes a BMC utility. The text portion of the statement identifies the invoked program name and the parameters that are passed to it. This program runs as a subtask. Figure 177 shows an example of the -BMCU command.
-DBUG—Debug

The -DBUG command turns on or off the DEBUG and FLOW options. The following keywords are valid:

- DEBUG
- DEBUGOFF
- FLOW
- FLOWOFF

**NOTE**
The utility command starts on the second line of the statement. The first line is reserved for parameters that are passed to the utility, such as NEW, TERM, MAINT, NEW/RESET, and TERM/RESET.

This command is not generated by ALTER or CHANGE MANAGER.

-DSN1—DSN1COPY (IBM utility)

The -DSN1 command invokes the IBM DSN1COPY utility. If you need to code this command yourself, use the following rules:

- The **Command** line (the one with the -DSN1 command on it) must contain only the sequence number and the command.
- The second statement should contain the DD keyword DSN1DDIN followed by a ddname.
- The third statement should contain the input DNAME for DSN1DDIN.
The fourth statement should contain the output DD keyword DSN1DDOU, followed by a ddname for the output data set that is defined in the JCL. If the output DSN begins with a period, the DSN is STOGROUP-defined, and the Execution component looks up the high level qualifier (Parm='parameter string').

Do not define a SYSUT1 DD in the JCL.

A -DSN1 command generated by CATALOG MANAGER executes as shown in Figure 178.

Figure 178  -DSN1 command

```
-DSN1 000950
   DSN1DDIN SYSR1000
   JFL.JFLTS24.COPY3
   DSN1DDOU DSN11000
   .DSNDBC.JFLDB06.JFLTS24A.I0001.A001
   PARM='OBIDXLAT,RESET,FULLCOPY'
   DBID 337,NEWDB=JFLDB06
   PSID 26,NEWTS=JFLTS24A
   OBID 28,NEWTB=JEFFZ.JFLTB242
   OBID 29,NEWTB=JEFFZ.JFLTB241
   OBID 27,NEWTB=JEFFZ.JFLTB243
   PART 0
```

For more information, see the documentation for the IBM utilities.

---

**NOTE**

CATALOG MANAGER and DASD MANAGER PLUS generate this command.

---

**-MERG—MERGECOPY (IBM utility)**

The -MERG command invokes the IBM DB2 MERGECOPY utility. This command and its parameters are passed to DB2. Figure 179 shows an example of the -MERG command.

Figure 179  -MERG command

```
-MERG 050000
   MERGECOPY TABLESPACE BMCASU.BMCUT01
   DEVT SYSDA
   COPYDDN SYSC0004
```
For more information, see the documentation for the IBM utilities.

**NOTE**
CATALOG MANAGER and DASD MANAGER PLUS generate this command.

---

**-MODI—MODIFY (IBM utility)**

The -MODI command invokes the IBM DB2 MODIFY RECOVERY or MODIFY STATISTICS utility. Figure 180 shows an example of the -MODI command for the MODIFY RECOVERY utility.

**Figure 180 -MODI command—MODIFY RECOVERY**

<table>
<thead>
<tr>
<th>-MODI 000001</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODI RECOVERY</td>
</tr>
<tr>
<td>TABLESPACE ACTQX18.ACTS0118</td>
</tr>
<tr>
<td>DELETE AGE(*)</td>
</tr>
</tbody>
</table>

Figure 181 shows an example of the -MODI command for the MODIFY STATISTICS utility.

**Figure 181 -MODI command—MODIFY STATISTICS**

<table>
<thead>
<tr>
<th>-MODI 000002</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODI STATISTICS</td>
</tr>
<tr>
<td>TABLESPACE ACTQX18.ACTS0118</td>
</tr>
<tr>
<td>DELETE ALL AGE(*)</td>
</tr>
</tbody>
</table>

For more information, see the documentation for the IBM utilities.

**NOTE**
CATALOG MANAGER and DASD MANAGER PLUS generate this command.

---

**-NOOP—No operation**

The -NOOP worklist command bypasses the statement that follows the command and proceeds to the next worklist command. Figure 182 shows an example of the -NOOP command.
The -QUI command invokes the IBM DB2 QUIESCE utility. Figure 183 shows an example of the -QUI command.

For more information, see the documentation for the IBM utilities.

NOTE

CATALOG MANAGER and DASD MANAGER PLUS generate this command.
The -REPO command invokes the IBM DB2 REPORT utility on table spaces or table space sets as required. The product passes this command and its parameters to DB2. Figure 184 shows an example of the -REPO command.

**Figure 184  -REPO command**

```
-REPO 000001
   REPORT RECOVERY TABLESPACE
   ACTQX16.ACTS0116
   CURRENT
   SUMMARY
   LOCALSITE
   RECOVERYSITE
```

For more information, see the documentation for the IBM utilities.

**NOTE**

CATALOG MANAGER and DASD MANAGER PLUS generate this command.

---

The -REPX command invokes the IBM DB2 REPAIR utility on table spaces as required. The product passes this command and its parameters to DB2.

For more information, see the documentation for the IBM utilities.

**NOTE**

The first line of the -REPX command must be blank.

DASD MANAGER PLUS generates this command.
–SQLM (Group multiple ALTER statements)

The –SQLM command (Figure 185) groups multiple ALTER statements (for table space partitions and their indexes) within a single command to reduce the total number of steps in the worklist and increase worklist performance.

Figure 185 –SQLM command

```
-SQLM 000002 ALTER TABLESPACE ARMDBJTR.TS40P5 PART 001
       PRIQTY      48 SECQTY     720;
       ALTER TABLESPACE ARMDBJTR.TS40P5 PART 002
       PRIQTY      48 SECQTY     720;
       ALTER TABLESPACE ARMDBJTR.TS40P5 PART 003
       PRIQTY      48 SECQTY     720;
       ALTER TABLESPACE ARMDBJTR.TS40P5 PART 004
       PRIQTY      48 SECQTY     720;
```

NOTE
DASD MANAGER PLUS generates this command.

-STOS—STOSPACE (IBM utility)

The -STOS command invokes the IBM DB2 STOSPACE utility. Figure 186 shows an example of the -STOS command.

Figure 186 -STOS command

```
-STOS 010000
       STOSPACE STOGROUP DEV070
```

For more information, see the documentation for the IBM utilities.

NOTE
CATALOG MANAGER and DASD MANAGER PLUS generate this command.
## Commands

This appendix presents the following topics:

- **CATALOG MANAGER commands** ........................................ 533
- **DB2 action commands** .................................................. 537
- **Utility commands** ....................................................... 540
- **Utility list commands** .................................................. 543
- **Statistics commands** ................................................... 544
- **List commands** ........................................................... 544
- **User commands** ......................................................... 549
- **Data browsing and editing commands** .............................. 549
  - Command-line commands .............................................. 549
  - Line commands .......................................................... 552

### 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 64 lists the commands.

#### Table 64 CATALOG MANAGER commands (part 1 of 5)

<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 64  CATALOG MANAGER commands (part 2 of 5)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLY</td>
<td>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.</td>
</tr>
<tr>
<td>AUDIT</td>
<td>displays rows from the CATALOG MANAGER Audit Log table</td>
</tr>
<tr>
<td>BATCH</td>
<td>generates JCL for a CATALOG MANAGER job to run in batch mode. Use BATCH when lengthy processing time makes running the job online undesirable.</td>
</tr>
<tr>
<td>BROWSE</td>
<td>invokes the CATALOG MANAGER data browsing feature to browse data in the selected table or view</td>
</tr>
<tr>
<td>CANCEL</td>
<td>terminates the current display without saving any changes that might have been made</td>
</tr>
<tr>
<td>CASCADE</td>
<td>displays the privileges that the REVOKE command would revoke for the specified user</td>
</tr>
<tr>
<td>CATALOGHELP</td>
<td>displays information about columns in the DB2 catalog tables</td>
</tr>
<tr>
<td>CLIPBOARD</td>
<td>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. CATALOG MANAGER saves changes in the CATALOG MANAGER SQL_Table, not in the clipboard.</td>
</tr>
<tr>
<td>CLIST</td>
<td>allows CATALOG MANAGER to execute a CLIST that is available but is not in the commands table</td>
</tr>
<tr>
<td>CMD</td>
<td>see COMMANDS</td>
</tr>
<tr>
<td>COMMANDS</td>
<td>lists the valid CATALOG MANAGER commands for the current list type. For example, different commands are listed for table spaces and columns.</td>
</tr>
<tr>
<td>COPYAUTHS</td>
<td>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</td>
</tr>
<tr>
<td>CUSTOMIZE</td>
<td>allows you to customize the CATALOG MANAGER Primary Menu panel to include a subset of the options</td>
</tr>
<tr>
<td>D</td>
<td>displays the catalog row for the selected object</td>
</tr>
<tr>
<td>DCL</td>
<td>generates GRANTs for explicit privileges that are held on an object or by a user</td>
</tr>
<tr>
<td>DDL</td>
<td>displays the DDL for the selected object</td>
</tr>
<tr>
<td>DES</td>
<td>displays a subset of the description that is provided by the DESCRIBE command for tables, databases, and DBRM packages</td>
</tr>
<tr>
<td>DESCRIBE</td>
<td>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.</td>
</tr>
</tbody>
</table>
## Table 64  CATALOG MANAGER commands (part 3 of 5)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOPTS</td>
<td>displays the installation options for this session of CATALOG MANAGER</td>
</tr>
<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></td>
<td>You can use the ENVIRONMENT command to verify the plans and modules that are in use.</td>
</tr>
<tr>
<td>EXPLAIN</td>
<td>displays rows from the <code>ownerName.PLAN_TABLE</code> 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></td>
<td>You can invoke the BMC SQL Explorer Analysis function to analyze the SQL.</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></td>
<td>CATALOG MANAGER options determine which dependent objects are included in the dependent DDL.</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></td>
<td>You must issue this command from the Command line.</td>
</tr>
<tr>
<td>JOIN</td>
<td>builds an SQL SELECT statement template for multiple tables</td>
</tr>
<tr>
<td></td>
<td>Enter JOIN on the first table or view in the list that you want to be included in the join. Enter an equal sign (=) on all other tables or views to be included.</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></td>
<td>LOGRBA also records this information in the CATALOG MANAGER Drop Recovery Log.</td>
</tr>
</tbody>
</table>
### CATALOG MANAGER commands (part 4 of 5)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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. You can save the information in the SQL_TABLE or a PDS.</td>
</tr>
<tr>
<td>OPTIONS</td>
<td>displays the CATALOG MANAGER options panels</td>
</tr>
<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>(DB2 Version 9 or earlier) converts a plan to use only packages. 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 Command line, prints the current list. When issued on a list, a DESCRIBE command is executed and the output is routed to the PRINT data set. 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>PROFILE</td>
<td>when issued on a list, displays the Profiles List panel, which lists all of the customized session profiles for CATALOG MANAGER. When issued on the Command line of the Utility Selections panel, PROFILE displays a list of user and site profiles for the selected utility type or types. When issued on the Command line of a DB2 command panel, PROFILE saves the DB2 command syntax to a DB2 command profile.</td>
</tr>
<tr>
<td>REFRESH</td>
<td>when issued on the Command 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>
</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 65 lists the DB2 action commands.

**Table 64**  CATALOG MANAGER commands (part 5 of 5)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>TSO POFRESET</td>
<td>resets all of the ISPF variables in the ISPF profile with the variables in the initial POF</td>
</tr>
<tr>
<td>TSO POFRESET POF(dataSetName(POFMember))</td>
<td>resets all of the ISPF variables in the ISPF profile with the variables in the specified initial or user POF name.</td>
</tr>
<tr>
<td>X</td>
<td>when entered in the <em>Cmd</em> column of a list line, excludes objects from being processed by the command that you specify on the <em>Command</em> line</td>
</tr>
<tr>
<td>XX</td>
<td>excludes all of the objects between the first XX command and the next XX command that you enter in the <em>Cmd</em> column of a list line from being processed by the command that you specify on the <em>Command</em> line</td>
</tr>
<tr>
<td>Xnnn</td>
<td>excludes the object in a list line and the objects in the following <em>nnn</em> list lines from being processed by the command that you specify on the <em>Command</em> line</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 65 lists the DB2 action commands.

**Table 65**  DB2 action commands (part 1 of 4)

<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>
</tbody>
</table>
### Table 65  DB2 action commands (part 2 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIND</td>
<td>displays the Bind panels, from which you can perform the following tasks:&lt;br&gt;- use the Explain option to indicate whether access path information is saved in ownerName.PLAN_TABLE&lt;br&gt;- 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&lt;br&gt;&lt;br&gt;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.&lt;br&gt;&lt;br&gt;For a native SQL procedure, the command generates a DSN BIND PACKAGE DEPLOY statement.&lt;br&gt;&lt;br&gt;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&lt;br&gt;&lt;br&gt;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&lt;br&gt;&lt;br&gt;Enter <strong>DB2COMMAND PROMPT</strong> to display the same panel as the DB2 Commands action from the CATALOG MANAGER Primary Menu.</td>
</tr>
<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 active threads for the current subsystem</td>
</tr>
<tr>
<td>DISDATABASE</td>
<td>displays the status of the selected database</td>
</tr>
</tbody>
</table>

---

538  CATALOG MANAGER for DB2 User Guide
Table 65 DB2 action commands (part 3 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| DROP | 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. |
| DSNZPARM | executes the IBM DSNWZP stored procedure and formats the information for display |
| EXCHANGE | exchanges data between two tables, one of which must be a clone of the other table |
| FREE | generates SQL to free active versions of plans or packages |
| FREE ALL ALL | 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. |
| FREE INACTIVE | generates SQL to free inactive versions of plans or packages in a list  

FREE INACT or FREE INACTIVE specifies the DB2 PLANMGMTSCOPE(INACTIVE) clause. |
| GRANT | generates SQL to grant privileges |
| INSERT | builds an SQL INSERT statement template for a table or view, and invokes the ISPF editor to allow you to customize the statement |
| LABEL | generates an SQL LABEL command for the selected object |
| QCONNECT | 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. |
| 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 |
| REFRESH | 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. |
| REGENERATE | 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. |
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 66 lists the utility commands.

Table 65  DB2 action commands (part 4 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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</td>
</tr>
<tr>
<td>STOP</td>
<td>executes the DB2 STOP DATABASE command</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 66  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 BMCCHECK IX</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>displays panels from which you can input options and generate BMC CHECK PLUS jobs with the CHECK INDEX command</td>
</tr>
<tr>
<td>BMCCOPY</td>
<td>displays panels from which you can input options and generate BMC COPY PLUS jobs</td>
</tr>
<tr>
<td>BMCCOPY INDEX BMCCOPY IX</td>
<td>displays panels from which you can input options and generate BMC COPY PLUS jobs</td>
</tr>
<tr>
<td>BMCCOPY IX</td>
<td>displays panels from which you can input options and generate BMC COPY PLUS jobs</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 jobs</td>
</tr>
<tr>
<td>BMCREBUILD</td>
<td>displays panels from which you can input options and generate BMC RECOVER PLUS jobs with the REBUILD INDEX command</td>
</tr>
<tr>
<td>BMCREBUILD INDEX BMCREBUILD IX</td>
<td>displays panels from which you can input options and generate BMC RECOVER PLUS jobs with the REBUILD INDEX command</td>
</tr>
</tbody>
</table>
### Table 66  Utility commands (part 2 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCRECOVER</td>
<td>displays panels from which you can input options and generate BMC RECOVER PLUS jobs</td>
</tr>
<tr>
<td>BMCRECOVER INDEX</td>
<td></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 PLUS jobs</td>
</tr>
<tr>
<td>BMCREORG INDEX</td>
<td></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 jobs</td>
</tr>
<tr>
<td>BMCSTATS INDEX</td>
<td></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></td>
<td>You can also use BMCUHIST as a list command.</td>
</tr>
<tr>
<td>BMCUNLOAD</td>
<td>displays panels from which you can input options and generate BMC UNLOAD PLUS jobs</td>
</tr>
<tr>
<td>BMCUTIL</td>
<td>executes the BMCUTIL command, and displays the 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></td>
<td>To limit the amount of information displayed by the command, include the userID.utilityID qualifier after the command.</td>
</tr>
<tr>
<td>CHECK</td>
<td>displays panels from which you can input options and generate IBM CHECK DATA jobs</td>
</tr>
<tr>
<td>CHKD</td>
<td></td>
</tr>
<tr>
<td>CHECK INDEX</td>
<td>displays panels from which you can input options and generate IBM CHECK INDEX jobs</td>
</tr>
<tr>
<td>CHECK IX</td>
<td></td>
</tr>
<tr>
<td>CHKI</td>
<td></td>
</tr>
<tr>
<td>COPY</td>
<td>displays panels from which you can input options and generate IBM COPY jobs</td>
</tr>
<tr>
<td>COPY INDEX</td>
<td></td>
</tr>
<tr>
<td>COPY IX</td>
<td></td>
</tr>
<tr>
<td>COPYTOCOPY</td>
<td>displays panels from which you can input options and generate IBM COPYTOCOPY jobs</td>
</tr>
<tr>
<td>COPYTOCOPY INDEX</td>
<td></td>
</tr>
<tr>
<td>COPYTOCOPY IX</td>
<td></td>
</tr>
<tr>
<td>DISUTIL</td>
<td>executes the DB2 DISPLAY UTILITY command, and displays the status of utility jobs</td>
</tr>
<tr>
<td></td>
<td>To limit the amount of information displayed by the command, include the userID.utilityID qualifier after the command.</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>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>
</tbody>
</table>

---

**BMCRECOVER INDEX**

Displays panels from which you can input options and generate BMC RECOVER PLUS jobs.

**BMCREORG INDEX**

Displays panels from which you can input options and generate BMC REORG PLUS jobs.

**BMCREORG IX**

Displays panels from which you can input options and generate BMC REORG PLUS jobs.

**BMCSTATS INDEX**

Displays panels from which you can input options and generate BMCSTATS jobs.

**BMCMSTATS IX**

BMCMSTATS is a component of the BMC DASD MANAGER PLUS product.

**BMCUHIST**

Displays information from the BMC Utility History table.

You can also use BMCUHIST as a list command.

**BMCUNLOAD**

Displays panels from which you can input options and generate BMC UNLOAD PLUS jobs.

**BMCUTIL**

Executes the BMCUTIL command, and displays the BMC utilities that have not completed or have not been terminated.

From this list, you can terminate a utility.

To limit the amount of information displayed by the command, include the userID.utilityID qualifier after the command.

**CHECK**

Displays panels from which you can input options and generate IBM CHECK DATA jobs.

**CHKD**

Displays panels from which you can input options and generate IBM CHECK INDEX jobs.

**CHECK INDEX**

Displays panels from which you can input options and generate IBM CHECK INDEX jobs.

**CHECK IX**

Displays panels from which you can input options and generate IBM CHECK INDEX jobs.

**CHKI**

Displays panels from which you can input options and generate IBM COPY jobs.

**COPY**

Displays panels from which you can input options and generate IBM COPY jobs.

**COPY INDEX**

Displays panels from which you can input options and generate IBM COPYTOCOPY jobs.

**COPY IX**

Displays panels from which you can input options and generate IBM COPYTOCOPY jobs.

**COPYTOCOPY**

Displays panels from which you can input options and generate IBM COPYTOCOPY jobs.

**COPYTOCOPY INDEX**

Displays panels from which you can input options and generate IBM COPYTOCOPY jobs.

**COPYTOCOPY IX**

Displays panels from which you can input options and generate IBM COPYTOCOPY jobs.

**DISUTIL**

Executes the DB2 DISPLAY UTILITY command, and displays the status of utility jobs.

To limit the amount of information displayed by the command, include the userID.utilityID qualifier after the command.

**DSN1COMP**

Displays panels from which you can input options for the IBM DSN1COMP utility.

**DSN1COPY**

Displays panels from which you can input options for the IBM DSN1COPY utility.

**EXCEPTIONS**

Invokes the BMC DASD MANAGER PLUS product to report exceptions on the object.

**EXEC**

Displays panels from which you can input options for the IBM EXEC SQL utility.
### Table 66  Utility commands (part 3 of 4)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 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 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 utility</td>
</tr>
<tr>
<td>RUNSTATS IX</td>
<td>displays panels from which you can input options for the IBM RUNSTATS 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></td>
<td>To limit the amount of information displayed by the command, include the <code>userId.utilityId</code> qualifier after the command.</td>
</tr>
<tr>
<td>SYNC</td>
<td>executes the SYNC command and displays the BMC utility sync tables</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>
Table 66 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>

Utility list commands

Table 66 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 66) on a list.

Table 67 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 <strong>Cmd</strong> column</td>
</tr>
<tr>
<td></td>
<td>You can also use the ? command as a keyword in the syntax for a DB2 command.</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 <strong>Command</strong> 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 68** lists the 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>DESTATISTICS</td>
<td>displays statistical information and catalog row information for the selected object</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>

List commands

CATALOG MANAGER provides the commands that generate and manipulate lists. In addition to these list commands, you can use the 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 69** lists all of the codes and describes the type of list produced.

<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 in 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>
</tbody>
</table>
### Table 69  List commands (part 2 of 5)

<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATW</td>
<td>lists time windows for running autonomic stored procedures in the SYSIBM.SYSAUTOTIMEWINDOWS 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>
<tr>
<td>BR</td>
<td>invokes the options for the CATALOG MANAGER data browsing function</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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>
</tbody>
</table>
### Table 69  List commands (part 3 of 5)

<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
<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>
</tbody>
</table>
| LI MX objectType qualifier | when issued from the Command line, displays a secondary list that shows multiple object types associated with certain source object types  
The object name must be fully qualified. |
| LIST                   | when entered in the Cmd column of a list line, produces a sublist qualified by the object name on the current list  
Using LIST from the Command line discards all current lists and creates a new level-one list.  
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. |
| LK                     | lists the limit keys |
| LL                     | lists the logical unit (LU) names for a location from the SYSIBM.LULIST table |
| LM                     | lists the limits for LUNAMES and MODENAMES from the SYSIBM.LUMODES table |
| LO                     | lists the accessible remote locations from the SYSIBM.LOCATIONS table |
## Table 69  List commands (part 4 of 5)

<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
<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>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.SYSRELS 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>
</tbody>
</table>
### Table 69  List commands (part 5 of 5)

<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SY</td>
<td>lists the synonyms</td>
</tr>
<tr>
<td>SY</td>
<td>lists the synonyms</td>
</tr>
<tr>
<td>SYNC</td>
<td>lists the columns from the BMCSYNC table for the BMC utilities</td>
</tr>
<tr>
<td></td>
<td>The BMCSYNC table contains information about the status of the objects that the currently executing utilities are accessing.</td>
</tr>
<tr>
<td></td>
<td>This command is valid only from a BMCUTIL list or from rows for BMC utilities in a STATUS list.</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>
<tr>
<td>TM</td>
<td>displays a mixed list of table space sets</td>
</tr>
<tr>
<td>TN</td>
<td>lists the type of object and the qualified table name</td>
</tr>
<tr>
<td></td>
<td>The product lists the following CATALOG MANAGER object types:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></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>XC</td>
<td>lists copies from the BMCXCOPY table and the cabinet copies for the BMC COPY PLUS for DB2 product</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>XSR</td>
<td>displays a mixed list of the DB2 for z/OS XML schema repository (XSR) tables that store XML schema documents from the SYSIBM.XSROBJECTS and related XSR catalog tables</td>
</tr>
<tr>
<td>XT</td>
<td>lists the auxiliary and base tables</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 70 lists the 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 71 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>
<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><strong>Note</strong>: 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>
</tbody>
</table>
### Table 71  Command-line commands for browsing or editing data (part 2 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>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>
</tbody>
</table>

In data editing mode, saves your changes and ends the edit session normally

<table>
<thead>
<tr>
<th>Command (short form)</th>
<th>Function E=Edit, B=Browse</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIND (F)</td>
<td>E, B</td>
<td>locates a character string that you specify in the command</td>
</tr>
</tbody>
</table>

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.
### Table 71 Command-line commands for browsing or editing data (part 3 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>FREEZE</td>
<td>E, B</td>
<td>anchors table columns while you edit or browse data. The columns remain visible when you scroll left and right.</td>
</tr>
<tr>
<td>HOME</td>
<td>E, B</td>
<td>displays the left-most column and first row of the table, and positions the cursor at the Command line.</td>
</tr>
</tbody>
</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. |
Table 71  Command-line commands for browsing or editing data (part 4 of 4)

<table>
<thead>
<tr>
<th>Command (short form)</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVE</td>
<td>E</td>
<td>saves your changes without ending the edit session</td>
</tr>
<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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In data editing mode, you can also modify the data.</td>
</tr>
</tbody>
</table>

**Line commands**

Table 72 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 72  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>Ia</td>
<td>inserts one or more blank lines below the line on which the command is entered</td>
</tr>
<tr>
<td>Da b</td>
<td>deletes one or more lines</td>
</tr>
<tr>
<td>Ra b</td>
<td>inserts a line with the same values as the line on which the command is entered</td>
</tr>
<tr>
<td>Ca b</td>
<td>identifies the source line of a copy operation</td>
</tr>
<tr>
<td>Ma 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 72  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&lt;sup&gt;a,b&lt;/sup&gt;</td>
<td>reverses the changes that you have made since the most recent SAVE</td>
</tr>
<tr>
<td>Z</td>
<td>displays the data in the row on which the command is entered vertically rather than horizontally, with one column per line</td>
</tr>
<tr>
<td>V</td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>This vertical row format is called <em>row view</em>. By using row view, you can display columns that are too large to display in column view.</td>
</tr>
<tr>
<td></td>
<td>To return the display to column view, press END.</td>
</tr>
</tbody>
</table>

<sup>a</sup> You can follow these commands with a numeric value to apply the command multiple times.

<sup>b</sup> 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.

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 553
$ACTCMD macro 149
$ACTEXC macro 153, 154
$ACTLISTC macro 153
$ACTULOG macro 153
$ACTVARS macro 153
$AJXDOC data set 380
&ACTCOLID ISPF variable 157
&ACTSRVR ISPF variable 157
&CLIST parameter 149
&CMD parameter 150
&DB2MAX parameter 150
&DB2MIN parameter 150
&HELP parameter 150
&LOAD parameter 150
&LOG parameter 150
&LSTO parameter 150
&NLIST parameter 150
&OBJECTS parameter 150
&NOSERVER parameter 151
&NUM parameter 151
&OBJECTS parameter 151
&SSID ISPF variable 157
&SYSTEM variable 419
&VCAT ISPF variable 157
&WFEK parameter 152
**PREFIX** TEMPLATE descriptor variable
 PREFIX symbolic variable 416
 SYSUID symbolic variable 418
 USERID symbolic variable 419
 ZPREFIX symbolic variable 421
 ZSYSID symbolic variable 421
 ZUSER symbolic variable 421
*DROP marker 282
*PERSIST option 81
+ command 543
+1 OUTPUT descriptor variable 411
+1 TEMPLATE descriptor variable 411
= command 543
=X command 46
? command 543

Numerics

10-byte object type address (TYPE) 155
16-byte function address (FUNC) 155
18-byte object name address (NAME) 155
2MEGSQL AEXIN keyword 403
2MEGSQL POF keyword 450
2SQL command 284
2WL command 533
8-byte object name qualifier (QUAL) 155

A

above-the-bar storage 70
ACM AEXIN keyword 403
ACM_AMS POF keyword 450
ACM_BASDIAG POF keyword 451
ACM_BRPNDIAG POF keyword 451
ACM_BRPTDSN POF keyword 451
ACM_CDLDSN POF keyword 451
ACM_CDLPS POF keyword 451
ACM_CDLSS POF keyword 451
ACM_CDLU POF keyword 451
ACM_CMPDIAG POF keyword 452
ACM_CPLCDLO POF keyword 452
ACM_CPLDIAG POF keyword 452
ACM_CPLWDSN POF keyword 452
ACM_CPLWDSNO POF keyword 452
ACM_DBRM1 POF keyword 452
ACM_DBRM2 POF keyword 452
ACM_DBRM3 POF keyword 452
ACM_DYNSORTW_NUM POF keyword 453
ACM_DYNSORTW_UNIT POF keyword 453
ACM_GLID POF keyword 453
ACM_IBMR_MAP_REQ POF keyword 453
ACM_IMPDIAG POF keyword 453
ACM_JDSN POF keyword 454
ACM_JDSNB POF keyword 454
ACM_JDSNBG POF keyword 454
ACM_JDSNC POF keyword 454
ACM_JDSNCPL POF keyword 455
ACM_JDSNCPL C++ keyword 455
ACM_JDSNCPL POF keyword 455
ACM_JDSNE POF keyword 455

Index 563
ACM_JDSNI POF keyword 455
ACM_PARALLEL_MAXINIT POF keyword 455
ACM_PARALLEL_MININIT POF keyword 456
ACM_PARALLEL_WORKLIST POF keyword 456
ACM_PARALLEL_XIMGROUPOF keyword 456
ACM_PARALLEL_XIMPST POF keyword 456
ACM_PARALLEL_XIMPROC POF keyword 456
ACM_PARALLEL_XIMSTRT POF keyword 456
ACM_PARALLEL_XIMTRCE POF keyword 457
ACM_PIC POF keyword 457
ACM_SDSN POF keyword 457
ACM_SDSNE POF keyword 457
ACM_WDSN POF keyword 457
ACM_WLORDER POF keyword 458
ACM_WLORDERMSG POF keyword 458
ACT product code 69
ACTCOMND member 144, 153
ACTCOMNU member 153
ACTIVATE command 537
ACTPSS CLIST 392
ACTUSER member 153
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 69
ACTvrDU Grant Authorities plan 69
ADDLOAD1 POF keyword 459
ADDLOAD2 POF keyword 459
ADMIN_COMMAND_DB2 stored procedure 164, 169, 170
ADMIN_DS_LIST stored procedure 164
Administrative Assistant for DB2 30
AJX4PART SLIB variable 416
AJX4YDDD SLIB variable
  DATEJ symbolic variable 410
  JDATE symbolic variable 412
  JU symbolic variable 414
  JU4Y symbolic variable 414
  YE symbolic variable 421
  YEAR symbolic variable 421
  YYYYDDD symbolic variable 421
AJX5PART SLIB variable 416
AJXBMCCP SLIB variable 410
AJXCR SLIB  variable
  CR symbolic variable 410
  IXCRR symbolic variable 412
AJXDB SLIB variable 410
AJXB2V2 SLIB variable 410
AJXB2V3 SLIB variable 410
AJXDDN SLIB variable 412
AJXDDOPT SLIB variable 411
AJXDDSQC SLIB variable
  DDSEQ symbolic variable 411
  SEQ symbolic variable 417
  SQ symbolic variable 417
AJXDSN44 SLIB variable 409
AJXFCCMD SLIB variable 411
AJXFJCHR SLIB variable 412
AJXGDGSLIB variable 411
AJXHM SLIB variable 411
AJXHMS SLIB variable
  HMS symbolic variable 411
  HO symbolic variable 411
  HOUR symbolic variable 411
  JHMS symbolic variable 412
  MI symbolic variable 415
  MINUTE symbolic variable 415
  SC symbolic variable 417
  SEC symbolic variable 417
  SECOND symbolic variable 417
  Ti symbolic variable 418
  TIME symbolic variable 418
  TIME4 symbolic variable 418
AJXIC SLIB variable 411
AJXIN input stream 379
AJIXL SLIB variable 412
AJIXXNOD SLIB variable 412
AJIXXSPC SLIB variable 412
AJXJAID SLIB variable 409
AJXJDDN SLIB variable 410
AJXJJULD SLIB variable 412
AJXJOBNM SLIB variable 413
AJXJOBNT SLIB variable 413
AJXPCOD SLIB variable 413
AJXQCID SLIB variable
  JQID symbolic variable 413
  WKID symbolic variable 420
  WORKID symbolic variable 420
AJXJSSID SLIB variable
  ATTACH symbolic variable 409
  GRPNM symbolic variable 411
  SS symbolic variable 417
AJXJYMD SLIB variable 414
AJXLDEFN SLIB variable 414
AJXLLQ SLIB variable 414
AJXR SLIB variable
  LOCREM symbolic variable 414
  LR symbolic variable 414
  TYPE symbolic variable 419
AJXMEMBER SLIB variable 415
AJXMEMBR SLIB variable 415
AJXMSID SLIB variable 415
AJXOBJT SLIB variable 416
AJXOBNAM SLIB variable 415
AJXOBNOD SLIB variable 415
AJXOBT SLIB variable 415
AUDIT installation option 427
audit logs
  administrative functions 374
  AUDIT_LOG table 157
audited events 374
AUDPOL installation option 434
authorization
  required for object creation 252
  required for SEARCH 184
  restricting access through plans 69
  setting 71
authorization object type 39
authorizations, verifying 348
AUTO command 316
AUTOALE installation option 434
AUTORUN installation option 434
AUTOWIN installation option 434
auxiliary table object type 39, 185
AUXRELS installation option 434

B
B command 228
B line command 552
BATCH
  CATALOG MANAGER lists 214
  HDDL output data set 212
  use with DDL, DESCRIBE, HDESCRIBE 209
  use with SEARCH 196
BATCH command 209, 534
BDSN installation option 427
BIND command 70, 538
BINDCOPY command 538
BINDDEPLOY command 538
BINDFAIL AEXIN keyword 403
BINDFAIL POF keyword 462
block QQ command 401
BLRP data set 114
BLRP_DATACLASS POF keyword 462
BLRP_DATACLASS_ALT POF keyword 462
BLRP_EXPDT POF keyword 462
BLRP_MGMTCLASS POF keyword 462
BLRP_MGMTCLASS_ALT POF keyword 463
BLRP_PREFIX POF keyword 463
BLRP_PRIQTY POF keyword 463
BLRP_RETPD POF keyword 463
BLRP_SECQTY POF keyword 463
BLRP_STACK POF keyword 463
BLRP_STORCLASS POF keyword 463
BLRP_STORCLASS_ALT POF keyword 463
BLRP_THRESH POF keyword 464
BLRP_UNIT POF keyword 464
BLRP_UNIT_ALT POF keyword 464

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 POF keyword 464
BMC_CHECK_OPTS AJXPOFIN keyword 404
BMC_CHECK_OPTS POF keyword 464
BMC_COPY_LOAD POF keyword 464
BMC_COPY_OPTS AJXPOFIN keyword 405
BMC_COPY_OPTS POF keyword 465
BMC_LOAD_LOAD POF keyword 465
BMC_LOAD_OPTS AJXPOFIN keyword 406
BMC_LOAD_OPTS POF keyword 465
BMC_RECOVER_LOAD POF keyword 465
BMC_RECOVER_OPTS AJXPOFIN keyword 407
BMC_RECOVER_OPTS POF keyword 466
BMC_REORG_LOAD POF keyword 466
BMC_REORG_OPTS AJXPOFIN keyword 407
BMC_REORG_OPTS POF keyword 466
BMC_UNLOAD_LOAD POF keyword 466
BMC_UNLOAD_OPTS AJXPOFIN keyword 408
BMC_UNLOAD_OPTS POF keyword 467
BMCACTvr.ATTR table 157
BMCACTvr.ATTR_VAL table 157
BMCACTvr.AUDIT_LOG table 157
BMCACTvr.CRS_VAL table 157
BMCACTvr.DLG table 157
BMCACTvr.DLG_ATTR table 157
BMCACTvr.EDITOR_USERS table 158
BMCACTvr.FCRS table 158
BMCACTvr.GMAP table 158
BMCACTvr.MSG table 158
BMCACTvr.RECOVERY_LOG table 158
BMCACTvr.SEARCH_VARS table 158
BMCACTvr.SESSION_LOG table 158
BMCACTvr.SQL_TABLE table 158
BMCACTvr.VIEW table 158
BMCDASM2 CLIST 63, 237
BMCEXPLORE (BMCEX) command 392
BMCEXPLORE command 540
BMCEXPLORE (BMCEX) command 392
BMCEXPLORE command 540
BMCDLOAD command 290, 540
BMCREBUILD command 290, 540
BMCREBUILD INDEX command 290, 540
BMCREBUILD IX command 290, 540
BMCRECOVER command 290, 541
BMCRECOVER INDEX command 290, 541
BMCRECOVER IX command 290, 541
BMCREORG command 290, 541
BMCREORG INDEX command 290, 541
BMCREORG IX command 290, 541
BMCSSTATS command 290, 541
BMCUHIST command 541
BMCULOAD command 290, 541
BMCUUTIL command 541
BOPTS installation option 428
BP command 545
BP object type 39
BPLAN installation option 428
BR command 147, 228, 545
BROWSE command 228, 534
browsing logs 369
buffer pool object type 39

C
C line command 552
C2 command 545
C2 object type 40, 185
CA command 545
CA object type 39
Cabinet copy object type 41
CANCEL command 75, 534, 549
CANCEL ISPF command 45
carriage return, hexadecimal format 45
CASCADE BATCH command 210
CASCADE command 534
CAT_LOAD POF keyword 467
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
categories, list of 533
installation option descriptions 427
installation options, list of 423
plans, list of 69
product code 69
switches 94, 175
tables 157
catalog statistics 203
CATALOGHELP command 534
CATAUDIT AEXIN keyword 404
CATAUDIT installation option 404
CATDOPT AEXIN keyword 404
CATTOP installation option 428
CATRECOV installation option 404
CATRECOVER AEXIN keyword 404, 406
CATSTATS command 203, 544
CATUTIL AEXIN keyword 404
CD command 545
CD object type 40, 185
CHANGE command 549
Character Field Max Width option field 83
characters
  nonprintable 44
  nonviewable 44
check 2 object type 40
CHECK command 291, 541
check constraint object type 185
check dependent object type 40, 185
CHECK INDEX command 291, 541
CHECK IX command 291, 541
check object type 39, 185
CHECK+_LOAD POF keyword 467, 504
CHECKDE installation option 434
CHECKDOPT POF keyword 467
CHECKOPT AEXIN keyword 404
CHECKS installation option 434
CHECKS2 installation option 434
CHGMAN_LOAD POF keyword 467
CHKD command 291, 541
CHKI command 291, 541
CI command 545
CI object type 39
CK command 545
CK object type 39, 185
CL command 545
CL object type 40
cleanup, data set 107
CLEANUP_RC POF keyword 468
CLIPBOARD command 333, 534
CLIST
  BMCDMF2 63, 237
  BMCD2 73, 162, 237
  sample 153
  variables 363
  writing user commands 153
CLIST command 534
CMD command 534
CMD_DISABLE command equivalent 148
CNTL_DATACLASS POF keyword 468
CNTL_EXPDT POF keyword 468
CNTL_MGMTCLASS POF keyword 468
CNTL_PREFIX POF keyword 468
CNTL_PRIQTY POF keyword 468
CNTL_RETDP POF keyword 468
CNTL_SECQTY POF keyword 468
CNTL_STORCLASS POF keyword 469
CNTL_UNIT POF keyword 469
CNTLMOUT_DSN POF keyword 469
CNTLMSCH_DSN POF keyword 469
CO command 545
CO object type 39, 182, 185
COLAUTH installation option 434
COLDISH installation option 434
COLDISS installation option 434
COLDIST installation option 434
collection object type 39
COLSTAT installation option 434
COLSTATS command 544
column authorization object type 39
column label object type 40
column object type 39, 185
COLUMNH installation option 434
COLUMNS installation option 434
combined lists
command format 181
excluding objects 183
generating 182
valid source objects 181
COMMAND (CMD) command 56
column name (exccomnd) 154
column program, writing 154
command prompts 311
command recognition character 38
command reference 56
command syntax 311
commands
abbreviating 58
adding 146
CATALOG MANAGER, list of 533
changing functionality 147
command-line, list of 549
data browsing 228
data browsing, list of 549
data editing 235
data editing, list of 549
disabling 146
editing the commands table 146
erasing leftover characters 59
executing 56
Fast Path Navigation 63
including multiple objects 59
invoking BMC Software utilities 290
invoking IBM DB2 utilities 291
ISPF 45
issuing from Cmd column 51, 59
issuing from Command line 37
issuing similar 58
line, list of 552
list, list of 544
listing 56
omitting parameters 58
parameters 58
replacing format 146
shortcuts 60
typing 59
user-written 148
utility list, list of 543
utility, list of 540
valid for object list 58
Wait-for-Enter (WFE) 59
COMMANDS command 534
commands table
$ACTCMD macro 149
ACTCOMND member 144
adding user-written commands 148
log option (excclg) 154
modifying 146
retaining from previous release 152
REXX entry 149
syntax and parameters 149
tailoring 352
COMMENT command 538
comment lines 525
Common area, viewing 67
Common Explain component
    integrating with CATALOG MANAGER 391
    compiling and linking user commands table 146
    compiling SLIBs 387
    Confirm SQL panel
        for extended SQL processing 223
        setting default values 89
        working with SQL 216
CONNECT (CON) command
    attaching to SSID 161
    connecting to specified SSID 166, 167, 173
    connect and attach functions, defined 159
    CONNECT BATCH command 210
    CONNECT command 538
    CONNECT RESET command 164
    connecting to a specified SSID 166
    connecting to an SSID, using a saved connection 172
    Connections Table 173
    CONSTDE installation option 434
    constraint dependent object type 40, 185
    CONTAB command 173
    CONTRL installation option 434
    control characters, using in object names 44
    CONTEXT installation option 434
    conventions, documentation 20
    COPY command 291, 541, 549
    COPY INDEX command 291, 541
    COPY installation option 434
    COPY IX command 291, 541
    COPY+_LOAD POF keyword 469
    COPYAUTHS command 325, 333, 534
    COPYDOPT POF keyword 469
copying data
    Copy Table Rows option 238
data compatibility 241
    overview 241
    using COPY command 241
    using Copy Table Rows option 245
copying user privileges by privilege type 333
COPYOPT AEXIN keyword 405
COPYTOCOPY command 291, 541
COPYTOCOPY INDEX command 291, 541
COPYTOCOPY IX command 291, 541
COUNT command 202, 538
CP command 545
CP object type 40, 185
CP object type, search qualifier 44
CPYEXP_DATACLASS POF keyword 469
CPYEXP_EXPDT POF keyword 469
CPYEXP_MGMTCLASS POF keyword 470
CPYEXP_PREFIX POF keyword 470
CPYEXP_RETPD POF keyword 470
CPYEXP_UNIT POF keyword 470
CR symbolic variable 410
CREATE command 538
creating an object hierarchy 272
creating and editing referential and unique constraints 264
creating objects, using DDL commands 271
creating tables
  copying table design 258
  creating and editing constraints 264
  defining columns 260
  using an existing object 258
creating utility profile data sets 293
CRS installation option 428
CRS option with session profiles 364
CRS_VAL table 157
CUP installation option 428
customer support 3
CUSTOMIZE (CU) command 353, 354, 361
CUSTOMIZE command 534
customizing CATALOG MANAGER
  authorization 352
  creating a session profile 358
  CRS option 364
  editing a tailored commands table 362
  initial list filters 359
  Primary Menu 354, 361
  types of customization 351
customizing object list display 198
CUT command 332
CX command 545
CX object type 40
CXA command 545
CXA object type 40
CXATTR installation option 434
CXAUTH installation option 434
CTX command 545
CTX object type 40
D
D (DELETE) line command 552
D (DESCRIBE/SELECT) command 207
D command 534, 543
DA symbolic variable 410
DA TEMPLATE descriptor variable 410
DASD_LOAD POF keyword 470
DASDDOPT AEXIN keyword 405
DASDDOPT POF keyword 471
data browsing commands, list of 549
data editing and browsing
  creating and editing data 238
  data locking 238
  invoking 228, 235
  no lock parameter 237
  row lock parameter 237
  setting options 235
  table lock parameter 236
data editing commands, list of 549
data locking 238
data set information, obtaining on a remote SSID 168
data set object type 40
data set sizing 379
data sets
  $AJXDOC 380
  cleanup 107
  DATAWORK 111
deleting automatically 107
discard 107
error 107
LOGSORT 111
map 107
naming conventions 87
options 88
permanent work 114
prefixes 116
punch 107
resolving names 117
sample VSAM object 106
setting default processing options 114
setting options 87
sizing 105
SORTOnnn 108
SORTOUT 108, 114
SORTPnnn 108
SORTWORK 111
SUTnnn 108
SYSCOPY 114
SYSDISC 114
SYSErr 114
SYSMAP 114
SYSPUNCH 114
SYSREC 114
SYSUT 114
SYSUTnnn 108
unload 107
data sets (continued)
  work 107
  WRKnnn 108
data type object type 40, 185
DATA_PACKER_LOAD POF keyword 471
DATABASES installation option 434
Database Administration for DB2 30
database object type 39, 185
DATABASESETSIZING POF keyword 471
DATATYP installation option 434
DATAWK_NBR POF keyword 471
DATAWK_UNIT POF keyword 471
DATAWORK data set 111
DATE OUTPUT descriptor variable
  DATE symbolic variable 410
  DT symbolic variable 411
  JYMD symbolic variable 414
  YMD symbolic variable 421
DATE symbolic variable 410
DATE TEMPLATE descriptor variable
  DATE symbolic variable 410
  JYMD symbolic variable 414
  YMD symbolic variable 421
DATEJ symbolic variable 410
DAY OUTPUT descriptor variable 410
DAY symbolic variable 410
DAY TEMPLATE descriptor variable 410
DB command 545
DB object type 39, 179, 182, 185
DB OUTPUT descriptor variable
  CR symbolic variable 410
  DB symbolic variable 410
  DBNAME symbolic variable 410
  IXCR symbolic variable 412
  TBCR symbolic variable 418
  TBCRE symbolic variable 418
  VCAT symbolic variable 420
DB symbolic variable 410
DB TEMPLATE descriptor variable
  CR symbolic variable 410
  DB symbolic variable 410
  DBNAME symbolic variable 410
  IXCR symbolic variable 412
  TBCR symbolic variable 418
  TBCRE symbolic variable 418
  VCAT symbolic variable 420
DB.IS TEMPLATE descriptor variable 412
DB..SN TEMPLATE descriptor variable 415
DB..TS TEMPLATE descriptor variable 418
DB.TS OUTPUT descriptor variable
  IXNODE symbolic variable 412
  OBNAM symbolic variable 415
  OBNOD symbolic variable 415
  TBNODE symbolic variable 418
DB2
  initialization parameters, viewing 67
  plan name (exclplan) 154
  security 70
  special registers, viewing 67
  subsystem attached to (excssid) 154
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 command
  command profiles 318
  PROFILE command 318
  specifying command syntax 316
DB2 commands
  issuing 311
  using command prompts 312
  using model commands 315
  using the PROFILE command 314
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
DB2 utilities
  DSN1COPY 526
  DB2COMMAND command 538
  DB2EXIT POF keyword 471
  DB2-identifiers 174
  DB2LOAD POF keyword 472
  DB2STMSGS AEXIN keyword 405
  DB2V2 symbolic variable 410
  DB2V3 symbolic variable 410
  DBAUTH installation option 434
  DBCS installation option 428
  DBNAME symbolic variable 410
  DBRM installation option 434
  DBRM object type 39, 185
  DBRM, explaining with Common Explain 394
  -DBUG worklist command 526
  DCL BATCH command 210
  DCL command 337, 534
  DCLGEN command 538
  DDDD symbolic variable 410
  DDL BATCH command 210
  DDL BATCH processing 210
  DDL command 271, 534
  DDNAME symbolic variable 410
  DDNAME symbolic variable 410
  DDNOPT symbolic variable 411
  DDSEQ symbolic variable 411
  DEBUGUNLD AEXIN keyword 405
  debugging, setting default processing options 121
defaults, setting product options
  data sets 114
  debugging 121
  generation data groups 119
  JCL jobcard 100
  JCL static data sets 104
  JCL STEPLIBs 103
  LISTDEF data set 130
  product options file 133
  SORTWORK data sets 111
  tapes 109
  TEMPLATE data set 130
  utility module names 124
defaults, setting user options
  basic options 78
  general 71, 75
  general options 81
  non-worklist JCL 127
  object use options 84
  online reruns 126
  options data set 76
  SHRLEVEL CHANGE 126
  user variables 131
defining an options data set 76
defining table columns 260
DELETE (DEL) command 191
DELETE command 538
deleting, data sets automatically 107
delimited identifiers 82
DEPEND installation option 434
dependency object type 40
DES BATCH command 210
DES command 207, 534
DESCRIBE BATCH command 210
DESCRIBE BATCH processing 210
DESCRIBE command 205, 207, 393, 395, 534
DESCRIPTION options 97
DEST command 544
DESTATISTICS (DEST) command 207
DESTATISTICS BATCH command 210
DESTATISTICS command 544
development aids 153
DIAG_MSGCLASS POF keyword 472
DIS command 538
disabling commands 146
DISC_DATACLASS POF keyword 473
DISC_DATACLASS_ALT POF keyword 473
DISC_EXPRPT POF keyword 473
DISC_MGMTCLASS POF keyword 473
DISC_MGMTCLASS_ALT POF keyword 473
DISC_PREFIX POF keyword 473
DISC_PRIQTY POF keyword 473
DISC_RETPD POF keyword 473
DISC_SECQTY POF keyword 473
DISC_THRQTY POF keyword 474
DISC_THRESHOLD POF keyword 474
discard data set 107
DISDATABASE command 538
DISP_ALLOW_POPUP POF keyword 474
DISP_AUTO_TAB POF keyword 475
DISP_LOCATION POF keyword 475
DISP_OMIT_CHAR POF keyword 475
DISP_STATS POF keyword 475
DISP_VAR_DBUG POF keyword 475
DISPLAY command 538
DISPLAY DB2 command 210
displaying truncated utility profile ID 299
DISTHREAD command 538
distinct type object type 40, 185
DISTSTATS command 544
DM command 545
DM object type 39, 185
documentation
  field-level Help 19, 46
  online Help 19
  related publications 19
DOPTS command 71, 535
DOWN command 550
DOWN ISPF command 45
DP command 545
DP object type 40
DPT installation option 429
DRO installation option 429
*DROP marker 282
DROP and DROP RECOVERY functions
  batch processing 287
  excluding indexes from recovery 285
  objects excluded from recovery 285
  recovering multiple indexes 288
  table spaces 278, 283
DROP command 278, 539
DROP IS switch 282
DROP NOT DONE message 282
Drop Recovery Log
  actions recorded 374
  recording simulated DROP 278
dropping pending changes 276
dropping table spaces 276
DROPRECOVERY command 535
DROPR command 283
DROPR_NOIC POF keyword 406, 475
DROPR_NOIC POF keyword 475
DS command 545
DS object type 40
-DSN1 worklist command 526
DSN1COMP command 291, 541
DSN1COPY command 291, 541
DSN1COPY utility 287, 526
DSNCHECK44 POF keyword 476
DSNEXIT library 67
DSNHDECP parameter 67
DSNTIAD_PLAN POF keyword 476
DSNUM OUTPUT descriptor variable 411
DSNUM symbolic variable 411
DSNWZP stored procedure 164, 539
DSNZPARM command 67, 539
DSNZPARM parameter 67
DT command 545
DT object type 40, 185
DT symbolic variable 411
DT TEMPLATE descriptor variable 411
dynamic SQL 184
DYNWORKUNIT AEXIN keyword 405

E

E command 235
ED command 147, 235, 307, 543, 545
EDIT command 235, 535, 550
editing the commands table 146
EDITOR_USERS table 158
electronic documentation 19
EN command 545
EN object type 40
END command 316
END command, creating and editing tables 550
END ISPFL command 45
enquiry character, hexadecimal format 45
ENTER ISPFL command 45
ENV AEXIN keyword 405
ENVIRON installation option 434
ENVIRONMENT (ENVI) command 67
ENVIRONMENT command 535
environment variables object type 40
EPLAN installation option 429
ERR_DATACLASS POF keyword 476
ERR_DATACLASS_ALT POF keyword 476
ERR_EXPRPT POF keyword 476
ERR_MGMTCLASS POF keyword 476
ERR_MGMTCLASS_ALT POF keyword 476
ERR_PREFX POF keyword 477
ERR_PRFQTY POF keyword 477
ERR_RETPD POF keyword 477
ERR_SECQTY POF keyword 477
ERR_STORCLASS POF keyword 477
ERR_STORCLASS_ALT POF keyword 477
ERR_THRESH POF keyword 477
ERR_UNIT POF keyword 477
ERR_UNIT_ALT POF keyword 478
error data set 107
ESC installation option 429
examples
SQLM command 531
excclog parameter 154
exccnum parameter 154
exccobjc parameter 154, 155
exccobjl parameter 154, 155
exccomnd parameter 154
exccsrc parameter 154
EXCEPTIONS command 541
EXCHANGE command 539
excluding objects from processing 62, 285
excl parameter 154
excrc parameter 154
excessid parameter 154
excessid parameter 154
exctssid parameter 154
EXEC command 291, 541
EXEC_LOAD POF keyword 478
evaluation, worklist
  cleanup job step for data sets 107
deleting data sets automatically 107
EXPLAIN command 535
extended SQL processing 222
external hexadecimal format 44

F

F1 (HELP) 45
F10 (LEFT) 45, 551
F11 (RIGHT) 46, 551
F2 (SPLIT) 46
F3 (END) 45, 550
F4 (ZOOM) 46, 257, 552
F5 (RFIND) 551
F7 (UP) 46
F8 (DOWN) 45
F9 (SWAP) 46
Fast Path Navigation 63, 237
FCMD symbolic variable 411
FCPY_DATACLASS POF keyword 478
FCPY_EXPDT POF keyword 478
FCPY_MGMTCLASS POF keyword 478
FCPY_PREFX POF keyword 478
FCPY_PRFQTY POF keyword 478
FCPY_RETPD POF keyword 479
FCPY_SECQTY POF keyword 479
FCPY_STACK POF keyword 479
FCPY_STORCLASS POF keyword 479
FCPY_SUPPRESS_SUFF POF keyword 116, 479
FCPY_UNIT POF keyword 479
FCRS table 158
FEW command 316
FIELDS installation option 434
file tailoring 380, 386
FILT_DATACLASS POF keyword 479
FILT_EXPDT POF keyword 479
FILT_MGMTCLASS POF keyword 480
FILT_PREFIX POF keyword 480
FILT_PRIQTY POF keyword 480
FILT_RETPD POF keyword 480
FILT_SECQTY POF keyword 480
FILT_STORCLASS POF keyword 480
FILT_UNIT POF keyword 480
FIND command 550
FK command 545
FK object type 40
FLOW AEXIN keyword 405
FN command 545
FN object type 40, 185
FO command 545
FO object type 40
FOREIGN installation option 435
foreign key object type 40
foreign key, data set sizing 382
form feed character, hexadecimal format 45
FP command 545
FP object type 40, 185
FREE ALL ALL command 539
FREE command 70, 539
FREE INACT command 539
FREE INACTIVE command 539
FREEZE command 551
FS command 545
FS object type 40, 185
FUNC parameter 155
function routine object type 40
function to execute on return (exrc) 154

G

GDG (generation data group)
setting default processing options 119
specifying 116
symbolic variable 116
using for data sets 116
GDG symbolic variable 116, 411
GDG_MODEL POF keyword 480
general options, setting 81
generating batch JCL 214
generating JCL by batch processing 209
generation data group. See GDG
GET command 393, 396, 535
GMAP table 158
GPLAN installation option 429
GRANT command 539
granting table privileges 323
GRPAT installation option 429
GRPNM symbolic variable 411

H

HASHFAIL AEXIN keyword 405
HASHFAIL POF keyword 480
HASHWARNRC AEXIN keyword 405
HASHWARNRC POF keyword 481
HC command 89, 535
HDDL
  AUTH switch 273
  batch processing 209
  including GRANT 273
HDDL BATCH command 210
HDDL command 272, 535
HDDL CONCAT command 272
HDDL output data set 212
HDESC command 209
HDESC (HDESC) command 209
HDESC BATCH command 210
HDESC BATCH processing 210
HDESC command 209, 535
HDIX installation option 429
HDPL installation option 429
HDSY installation option 430
HDTB installation option 430
HDTR installation option 430
HDTLS installation option 430
HDVW installation option 430
HELP ISPF command 45
Help, online 19
hexadecimal format, null character 45
Hierarchical GRANT (HGRANT) command 328
HISTORY command 544
HLQ.UDBCNTL members
  product options file 439
HM symbolic variable 411
HMS symbolic variable 411
HO symbolic variable 411
HO TEMPLATE descriptor variable 411
HO.MI TEMPLATE descriptor variable
  HM symbolic variable 411
  TIME4 symbolic variable 418
host variables
  in SQL model statements 219
  long values 193
  testing 222
  used in SEARCH 191
  using quotation marks 193
  valid values in DML statements 193
HOUR OUTPUT descriptor variable 411
HOUR symbolic variable 411
HOUR.MINUTE OUTPUT descriptor variable
  HM symbolic variable 411
  TIME4 symbolic variable 418
HPLAN installation option 430
HPRINT command 535

I
I line command 552
IBM DB2 utilities available in CATALOG MANAGER 291
IBM utilities
  DSN1COPY 526
IC command 545
IC object type 40, 182, 185
IC symbolic variable 411
IC TEMPLATE descriptor variable
  IC symbolic variable 411
  ICTYPE symbolic variable 411
  JOBTYP symbolic variable 413
  OBJT symbolic variable 415
  OBJTYP symbolic variable 415
  RTYPE symbolic variable 416
  RUNTPY symbolic variable 417
  TYPE symbolic variable 419
ICCOL installation option 430
ICSYC installation option 430
ICTYPE OUTPUT descriptor variable 411
ICTYPE symbolic variable 411
ICTYPE TEMPLATE descriptor variable 411
identity column object type 41
identity columns 263
IEFUSI exit 70
IL command 545
IL object type 40
IM (data editing) command 551
IM command 546
IM object type 40, 182
image copy object type 40, 185
IMPORT command 535
IN command 546
IN object type 40, 185
INCLUDE_SYSPRIN2 POF keyword 481
including SEARCH in BATCH jobs 196
index and table space partition object type 41
index mixed object type 40
index object type 39, 185
index space object type 40, 185
index space partition object type 40, 44, 185
index space statistics object type 40
INDEXES installation option 435
indexes, recovering multiple 288
INDEXH installation option 435
INDEXPA installation option 435
INDEXPH installation option 435
INDEXSH installation option 435
INDEXST installation option 435
INDEXSTATS command 544
initial list filters
  creating 359
  retrieving 363
initial POF, refreshing 136
input streams 72
INSERT command 539
installation options
  CATALOG MANAGER 423
installation requirements 34
internal format 44
INVALID COMMAND message 401
IOLOAD1 POF keyword 481
IOLOAD2 POF keyword 482
IP command 546
IP list object type 40
IP name object type 40, 185
IP object type 40, 44, 185
IPLIST installation option 435
IPNAMES installation option 435
IS command 546
IS object type 40, 185
IS symbolic variable 412
IS TEMPLATE descriptor variable 412
ISPF
  AJXPODAT macro 135
  commands 45
  commands in CATALOG MANAGER 260
  file tailoring 386
  file tailoring for JCL generation 380
  file tailoring, used by JCL Generation 380
  help 45
  sharepool variables 153
  variables 157
  variables in data set names 87
ISS command 546
ISS object type 40
ITERATIONMODE AEXIN keyword 405
IX (data editing) command 551
IX command 546
IX object type 39, 182, 185
IX symbolic variable 412
IXCR symbolic variable 412
IXNAME symbolic variable 412
IXNODE symbolic variable 412
IXSPC symbolic variable 412
J
jar contents object type 40
jar object type 40
JARCONT installation option 435
JAROBJT installation option 435
Java archive contents object type 185
Java archive object type 185
Java option object type 40
Java path object type 40
JAVAPTH installation option 435
JAVOPTS installation option 435
JB command 546
JB object type 40, 185
JC command 546
JC object type 40, 185
JCL Generation
  data set sizing 381
  ISPF file tailoring 380
  product options 72
  RUNSTATS utility 381
  setting options 85
  VSAM object sampling 381
JCL job card, setting default processing options 100
JCL job statement, defaults 101
JCL static data sets, setting default processing options 104
JCL STEPLIBs, setting default processing options 103
JCLCLEANUP POF keyword 482
JCLLIB POF keyword 482
JD symbolic variable 412
JDATE OUTPUT descriptor variable
  DATEJ symbolic variable 410
  JDATE symbolic variable 412
  JU symbolic variable 414
  JUL4Y symbolic variable 414
  JULIAN symbolic variable 414
  YYYYYDDD symbolic variable 421
  YYYYYYDDD symbolic variable 421
JDATE symbolic variable 412
JDATE TEMPLATE descriptor variable
  DATEJ symbolic variable 410
  JDATE symbolic variable 412
  JU symbolic variable 414
  JUL4Y symbolic variable 414
  JULIAN symbolic variable 414
  YYYYYDDD symbolic variable 421
  YYYYYYDDD symbolic variable 421
JDAY OUTPUT descriptor variable
  DDD symbolic variable 410
  JDAY symbolic variable 412
  JJULD symbolic variable 412
JDAY symbolic variable 412
JDAY TEMPLATE descriptor variable
  DDD symbolic variable 410
  JDAY symbolic variable 412
  JJULD symbolic variable 412
JDDN symbolic variable 412
JDSN installation option 430
JESS POF keyword 482
JHMS symbolic variable 412
JJULD symbolic variable 412
JOB_INCLUDE_MEMBER POF keyword 482
JOBCHAR symbolic variable 412
JOBNAME OUTPUT descriptor variable
  JOBCHAR symbolic variable 412
  JOBNAME symbolic variable 413
  JPCOD symbolic variable 413
  MEMBER symbolic variable 415
  MEMBR symbolic variable 415
  PGMR symbolic variable 416
  WKOWN symbolic variable 420
  WKOWNER symbolic variable 420
  JOBNAME symbolic variable 413
  JOBNAME TEMPLATE descriptor variable
    ATTACH symbolic variable 409
    DDOP0T symbolic variable 411
    JDDN symbolic variable 412
    JOBOCHAR symbolic variable 412
    JOBNAME symbolic variable 413
    JPCOD symbolic variable 413
    MEMBER symbolic variable 415
    MEMBR symbolic variable 415
    PGMR symbolic variable 416
    WKOWN symbolic variable 420
    WKOWNER symbolic variable 420
    ZACCTNUM symbolic variable 421
JOBTYP symbolic variable 413
JOIN command 535
JP command 546
JP object type 40
JPCOD symbolic variable 413
JQID symbolic variable 413
JS1 symbolic variable 413
JS2 symbolic variable 413
JS4 symbolic variable 413
JSSID symbolic variable 413
JT command 546
JJT object type 40
JU symbolic variable 414
JUL4Y symbolic variable 414
JULIAN symbolic variable 414
JYMD symbolic variable 414
K
KC (data editing) command 551
KC command 546
KC object type 40
KCOLUSE installation option 435
key column object type 40
key column use object type 185
key column user object type 40
KEYS installation option 435
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
LOCATIO installation option 435
location list, connecting to a remote SSID 169
location object type 39, 185
lock contention during data editing 238
locking options 63, 237
LOCREM symbolic variable 414
LOCREM TEMPLATE descriptor variable 414
log text pointer (pointer) 155
LOGRBA command 535
logs
accessing 368
administering audit logs 374
administrative functions 368
browsing 369
DDL Audit Log 374
Drop Recovery Log 374
enable session 150
purging 372
Session Log 369, 370
types maintained 367
LOGSORT data set 111
LOGWK_NBR POF keyword 485
LOGWK_UNIT POF keyword 485
long names, truncating 83
LPART OUTPUT descriptor variable 414
LPART symbolic variable 414
LPLAN installation option 431
LR symbolic variable 414
LR TEMPLATE descriptor variable 414
LS command 547
LS object type 41, 185
LU command 547
LU list object type 40, 185
LU mode object type 40, 185
LU mode select object type 41, 185
LU name object type 41, 185
LU object type 41, 185
LULIST installation option 435
LUMODES installation option 435
LUNAMES installation option 435
MAP_MGMTCLASS_ALT POF keyword 485
MAP_MGMTCLASS POF keyword 485
MAP_DATACLASS POF keyword 485
MAP_DATACLASS_ALT POF keyword 485
MAP_EXPDT POF keyword 485
MAP_MGMTCLASS POF keyword 485
MAP_UNIT POF keyword 485
MAP_UNIT_ALT POF keyword 485
materialized query table object type 41
MAX installation option 431
MAX_CYL POF keyword 487
MAX_PRIQTY POF keyword 487
MAX_SECQTY POF keyword 487
MAX_UNITCNT POF keyword 487
MDDL command 273, 536
MEMBER symbolic variable 415
MEMBR symbolic variable 415
MEMLIMIT POF keyword 487
MEMLIMIT system parameter 70
-MERG worklist command 527
MERGECOPY command 291, 542
MERGECOPY worklist command 527
message files 137
message, invalid command 401
MI symbolic variable 415
MI TEMPLATE descriptor variable 415
MINUTE OUTPUT descriptor variable 415
MINUTE symbol variable 415
MINUTE TEMPLATE descriptor variable 415
mixed object lists
excluding objects 181
generating 179
valid source objects 178
mixed object type 41
MK command 547
MMDD symbolic variable 415
MO symbolic variable 415
MO TEMPLATE descriptor variable 415
MONTH OUTPUT descriptor variable 415
MONTH symbol variable 415
MONTH TEMPLATE descriptor variable 415
model commands 311
models
creating new objects 59
creating tables 258
MODESEL installation option 435
-MODI worklist command 528
MODIFY command 291, 542
MODIFY RECOVERY worklist command 528
MODIFY STATISTICS worklist command 528
MODIFYRECOVERY command 291, 542
MODIFYSTATISTICS command 542
MODELYSTATS command 291
MONTH OUTPUT descriptor variable 415
MONTH symbol variable 415
MONTH TEMPLATE descriptor variable 415
MONTH.DAY OUTPUT descriptor variable 415
MONTH.DAY TEMPLATE descriptor variable 415

M
M line command 552
macros
$ACTEXC 153, 154
$ACTULOG 153, 154
AJXPĐAT 135
MAINT command 372
MAINTAIN command 368, 372, 536
MANY command 316
map data set 107
MAP_DATACLASS POF keyword 485
MAP_DATACLASS_ALT POF keyword 485
MAP_EXPDT POF keyword 485
MAP_MGMTCLASS POF keyword 485

Index 577
MORE command 240, 551
MPLAN installation option 431
MQT command 547
MQT object type 41
MSG table 158
MSSID symbolic variable 415
multitasking, using tape stacking 111
MX command 179, 547
MX object type 41

N
name fields in user-written commands 156
NAME parameter 155
native SQL procedure object type 41, 185
navigating CATALOG MANAGER 47
negative acknowledge, hexadecimal format 45
No operation worklist command 528
NOAPFOK AEXIN keyword 406
NOFAILNOIMAGECOPY AEXIN keyword 406
NOLOADCOMP AEXIN keyword 406
NONE command 316
nonprintable characters 44
nonviewable characters 44
non-worklist JCL 127
-NOOP worklist command 528
NOSQLCOMP AEXIN keyword 406
NOSTARTOVER AEXIN keyword 406, 407
NOTIFYUNLD AEXIN keyword 406
NOWKIDREPLACE AEXIN keyword 406
NP command 547
NP object type 41, 185
null characters 44
number of objects option (exccnum) 154

O
OB command 547
OB object type 41
OBDS installation option 435
object lists
  customizing display 198
  excluding objects 62
  from different source object types 54
  from multiple source objects 52
  generating from Primary Menu 49
  reordering columns 198
  secondary lists 49
  selecting objects 61
  sorting by column 201
object role dependency object type 41
object type in command text (exccobjc) 154
object types
  generating lists 38
  passing in user-written commands 156
OBJROF installation option 435
OBJT symbolic variable 415
OBJTYP symbolic variable 415
OBNAME symbolic variable 415
OBNOD symbolic variable 415
online command reference 56
online Help 19, 46
online reorgs 126
online schema changes object type 41
OPT command (data editing and browsing) 551
OPT parameter 155
options
  changing 70
  data editing and browsing 228
  data set names 87, 88
  data sets 114
  debugging and display 121
  DESCRIBE 97
  general 81
  generation data group 119
  installation 71
  JCL Generation 85
  JCL job cards 100
  JCL static data sets 104
  JCL STEPLIBs 103
  LISTDEF data set 130
  non-worklist JCL 127
  online reorg 126
  option switches 94, 175
  panel attributes 93
  product switches 133
  refreshing 72
  setting 70
  setting values 75
  SHRLEVEL CHANGE 126
  SORTWORK data sets 111
  SQL command 89
  SQL SELECT 92
  tapes 109
  TEMPLATE data set 130
  used by products 74
  user 72
  user variables 131
  utility module names 124
OPTIONS command 536
options data set, defining 76
ORDER command 536
ordinary identifiers 82
ORTPARM_DSN POF keyword 487
OS command 547
OS object type 41
OUTPUT descriptor variables, list of 409
P command 543
PA command 547
PA object type 41
PA symbolic variable 416
PA TEMPLATE descriptor variable 416
PACKAGE installation option 435
package object type 39, 185
packages
explaining statements with Common Explain 395
explaining with Common Explain 394
use by plans 182
PACKAUT installation option 435
PACKCPY installation option 435
PACKIT command 70, 536
PACKLIS installation option 435
packlist object type 41
packlists 182
PACKSTM installation option 435
panel attribute options, setting 93
parameter object type 185
parameters
MEMLIMIT 70
system 70
PARMS installation option 435
parsing for object name 155
parsing object types and names 155
PART OUTPUT descriptor variable
BMCCP symbolic variable 410
PA symbolic variable 416
PART symbolic variable 416
PART4 symbolic variable 416
PART5 symbolic variable 416
UP symbolic variable 419
UPART symbolic variable 419
PART symbolic variable 416
PART TEMPLATE descriptor variable
BMCCP symbolic variable 410
DSNUM symbolic variable 411
LPART symbolic variable 414
PART symbolic variable 416
PART4 symbolic variable 416
PART5 symbolic variable 416
UP symbolic variable 419
UPART symbolic variable 419
PART4 symbolic variable 416
PART5 symbolic variable 416
PASTE command 333
PB symbolic variable 416
PB TEMPLATE descriptor variable 416
PCPY1_DATACLASS POF keyword 487
PCPY1_DATACLASS_ALT POF keyword 487
PCPY1_EXPDT POF keyword 487
PCPY1_MGMTCLASS POF keyword 488
PCPY1_MGMTCLASS_ALT POF keyword 488
PCPY1_PREFIX POF keyword 488
PCPY1_PRIQTY POF keyword 488
PCPY1_RETPD POF keyword 488
PCPY1_SECQTY POF keyword 488
PCPY1_STACK POF keyword 488
PCPY1_SUPPRESS_SUFF POF keyword 116, 489
PCPY1.THRESH POF keyword 489
PCPY1_UNIT POF keyword 489
PCPY1_UNIT_ALT POF keyword 489
PCPY2_DATACLASS POF keyword 489
PCPY2_DATACLASS_ALT POF keyword 489
PCPY2_EXPDT POF keyword 489
PCPY2_MGMTCLASS POF keyword 489
PCPY2_MGMTCLASS_ALT POF keyword 490
PCPY2_PRIQTY POF keyword 490
PCPY2.RETPD POF keyword 490
PCPY2_SECQTY POF keyword 490
PCPY2_STACK POF keyword 490
PCPY2_SUPPRESS_SUFF POF keyword 116, 490
PCPY2.THRESH POF keyword 491
PCPY2_UNIT POF keyword 491
PCPY2_UNIT_ALT POF keyword 491
PDD command 547
PDISTSTATS command 544
PDSN installation option 431
PEEK command 67
PENDDDL installation option 435
pending changes, dropping 276
percent character, hexadecimal format 45
permanent work data sets 114
*PERSIST option 81
PFSHOW ISPF command 46
PG command 547
PG object type 39, 182, 185
PGC command 547
PGMR symbolic variable 416
PI command 182, 547
PI object type 41
PK command 547
PK object type 41
PKSYSTE installation option 436
PL command 547
PL object type 39, 179, 182, 185
plan authorization object type 41
PLAN installation option 436
plan object type 39, 185
PLANAUT installation option 436
PLANDEP installation option 436
plans
CATALOG MANAGER 69
explaining with Common Explain 394
plans (continued)  
manipulating 70  
package use 182  
PLP installation option 431  
PLSYSSTE installation option 436  
PM command 547  
POF (product options file)  
adding steps to JCL 141  
AJXPODAT macro 135  
creating a user POF 133  
description 439  
generating reports 137  
initial 72  
initializing 72  
keyword descriptions 450  
keywords, list of 439  
obtaining list of TEMPLATEs or LISTDEFs 144  
overriding values in SLIBs 139  
POF Validation Report 137  
populating 72  
refreshing the initial POF 136  
reusing in a subsequent installation 138  
sample file 439  
setting default processing options 133  
updating a user POF 134  
user 73  
using multiple POFs 135  
Variables Initialized with Default report 137  
POFDATE POF keyword 73, 491  
POFDS installation option 72, 432, 439  
pointer parameter 155  
PR command 547  
PR object type 39, 185  
PR parameter in ACTEMAIN CLIST 363, 364  
PRE_JOBSTEP_INCLUDE POF keyword 491  
PREFIX symbolic variable 416  
prefixes 116  
PRI command 209  
PRIBAC symbolic variable 416  
PRIBAC TEMPLATE descriptor variable 416  
primary key object type 41  
PRINT CLOSE command 89  
PRINT command 89, 209, 536  
privileges  

copying by privilege type 333  
granting table privileges 323  
to create objects 252  
privileges, reassigning 348  
privileges, revoking 348  
PRO command 536  
PROC_BMCLOAD_STEP POF keyword 492  
PROC_BMCLOAD_STEP POF keyword 492  
PROC_BMCCHECK_NAME POF keyword 491  
PROC_BMCCHECK_STEP POF keyword 491  
PROC_BMCCOPY_NAME POF keyword 492  
PROC_BMCCOPY_STEP POF keyword 492  
PROC_BMCPPRS_NAME POF keyword 492  
PROC_BMCPPRS_STEP POF keyword 492  
PROC_BMCLOAD_NAME POF keyword 492  
PROC_BMCRECOVER_NAME POF keyword 493  
PROC_BMCRECOVER_STEP POF keyword 493  
PROC_BMCREORG_NAME POF keyword 493  
PROC_BMCREORG_STEP POF keyword 493  
PROC_BMCSSTATS_NAME POF keyword 493  
PROC_BMCSSTATS_STEP POF keyword 493  
PROC_BMCSTOP_NAME POF keyword 494  
PROC_BMCSTOP_STEP POF keyword 494  
PROC_BMCTTRIG_NAME POF keyword 494  
PROC_BMCTTRIG_STEP POF keyword 494  
PROC_BMCUNLOAD_NAME POF keyword 494  
PROC_BMCUNLOAD_STEP POF keyword 495  
PROC_BMCUPRS_NAME POF keyword 495  
PROC_BMCUPRS_STEP POF keyword 495  
PROC_DSN1COPY_NAME POF keyword 496  
PROC_DSN1COPY_STEP POF keyword 496  
PROC_DSNUTILB_NAME POF keyword 495  
PROC_DSNUTILB_STEP POF keyword 495  
PROC_GEN_SET_VAR POF keyword 496  
PROC_IDCAM_NAME POF keyword 496  
PROC_IDCAM_STEP POF keyword 496  
PROC_IEFFB14_NAME POF keyword 496  
PROC_IEFFB14_STEP POF keyword 497  
PROC_TS_NAME POF keyword 497  
PROC_TS_STEP POF keyword 497  
PROC_USE POF keyword 497  
PROC_USER_DEF_STEP POF keyword 497  
PROC_USER_DEFINED POF keyword 497  
PROCEDU installation option 436  
procedure object type 185  
product Help 19, 46  
product options file. See POF  
product support 3  
PROFILE command 314, 318, 536  
PROFILE command (session profiles) 356  
PROFILE command (utility profiles) 306  
PROFILE SAVE command 304  
PROFILE SAVEAS command 298, 301, 303  
PROFILE SET command 306  
PROFILE.profileName ADDED message 355, 359, 360  
PROFILE.profileName UPDATE message 362  
PROFILES command 536  
PT command 547  
PT object type 41, 182  
publications, related 19  
punch data set 107  
PUNCH_DATACLASS POF keyword 498  
PUNCH_EXPDT POF keyword 498  
PUNCH_MGMTCLASS POF keyword 498  
PUNCH_PREFIX POF keyword 498  
PUNCH_PRIQTY POF keyword 498  
PUNCH_SECQTY POF keyword 498  
PUNCH_STORCLASS POF keyword 498  
PUNCH_UNIT POF keyword 499  
PURGE COMPLETED message 373  
purging logs 372
Q

QCONNECT command 539
QQ command 401
QRO command 547
QRP command 547
QRY command 547
QUAL parameter 155
Qualifier field
  IP object type 44
  object names 42
  objects with two-part names 44
  TP object type 44
  wildcard characters 42
QUEROPT installation option 436
QUERPLN installation option 436
QUERY installation option 436
-QUI worklist command 529
quickname 172
quick-search
  in BATCH jobs 196
  using saved search variables 194
  WHERE clause 195
QUIESCE command 291, 542
QUIESCE utility worklist command 529

R

R line command 552
RC parameter 155
RCCOL installation option 432
RCHANGE command 551
RCPI1_DATACLASS POF keyword 499
RCPI1_DATACLASS_ALT POF keyword 499
RCPI1_EXPD T POF keyword 499
RCPI1_MGMTCLASS POF keyword 499
RCPI1_MGMTCLASS_ALT POF keyword 499
RCPI1_PREFIX POF keyword 499
RCPI1_PRIQTY POF keyword 499
RCPI1_RETPD POF keyword 499
RCPI1_SECQTY POF keyword 500
RCPI1_STACK POF keyword 500
RCPI1_STORCLASS POF keyword 500
RCPI1_STORCLASS_ALT POF keyword 500
RCPI1_SUPPRESS_SUFF POF keyword 116, 500
RCPI1_THRESH POF keyword 500
RCPI1_UNIT POF keyword 500
RCPI1_UNIT_ALT POF keyword 500
RCPI2_DATACLASS POF keyword 501
RCPI2_DATACLASS_ALT POF keyword 501
RCPI2_EXPD T POF keyword 501
RCPI2_MGMTCLASS POF keyword 501
RCPI2_MGMTCLASS_ALT POF keyword 501
RCPI2_PREFIX POF keyword 501
RCPI2_PRIQTY POF keyword 501
RCPI2_RETPD POF keyword 501
RCPI2_SECQTY POF keyword 501
RCPI2_STACK POF keyword 500
RCPI2_STORCLASS POF keyword 500
RCPI2_STORCLASS_ALT POF keyword 500
RCPI2_SUPPRESS_SUFF POF keyword 116, 502
RCPI2_THRESH POF keyword 502
RCPI2_UNIT POF keyword 502
RCPI2_UNIT_ALT POF keyword 502
RD command 547
RD object type 41
RE command 547
RE object type 41, 182, 185
real-time statistics. See RTS
REBIND command 70, 539
REBINDFAIL AEXIN keyword 406
REBINDFAIL POF keyword 503
REBINDRC AEXIN keyword 407
REBINDRC POF keyword 503
REBUILD command 291, 542
REBUILD INDEX command 291, 542
REBUILD IX command 291, 542
RECOVER command 287, 288, 291, 542
RECOVER INDEX command 291, 542
RECOVER IX command 291, 542
RECOVER+ LOAD POF keyword 503
RECOVERDOPT POF keyword 503
recovery objects 278
RECOVEROPT AEXIN keyword 407
RECOVERY_LOG table 158
referential integrity object type 41
REFRESH command 536, 539
refreshing user options 72
REGENERATE command 539
REGION POF keyword 503
register for return code (RC) 155
relation object type 41, 185
RELS installation option 436
remote SSID, connecting 168, 169
RENAME command 540
reordering object list columns 198
REORG command 291, 542
REORG INDEX command 291, 542
REORG IX command 291, 542
REORG+ LOAD POF keyword 504
REORG_MAPTAB POF keyword 503
REORGDOPT POF keyword 504
REORGOPT AEXIN keyword 407
REPAIR utility worklist command 530
-REPO worklist command 530
REPORT command 292, 542
REPORT INDEX command 292, 542
REPORT IX command 292, 542
REPORT utility worklist command 530
reports, generating POF 137

Index  581
S

S command 536
SAVE command 552
SC command 547
SC object type 41
SC symbolic variable 417
SC TEMPLATE descriptor variable 417
SCHEMAA installation option 436
SE command 547
SE object type 41, 185
SEARCH command 184, 536
SEARCH function
  associate with profile 189
  associate with user ID 189
  authorization required 184
  complex subqueries 196
  generating lists 186
  JOINed tables 197
  operator (Oper) variables 187
  retrieving saved variables 190
  saving variables 188
  setting values 188
  using host variables 191
  valid source objects 185
WHERE clauses 188, 192
SEARCH_VARS table 158
SEC OUTPUT descriptor variable 417
SEC symbolic variable 417
SECOND OUTPUT descriptor variable 417
SECOND symbolic variable 417
SECOND TEMPLATE descriptor variable 417
secondary lists, generating from object lists 50
security, setting authorizations 71
SEE command 67, 540
SELECT command 540
SEQ OUTPUT descriptor variable 418
DDPOT symbolic variable 411
DDSEQ symbolic variable 411
JDDN symbolic variable 412
RSEQ# symbolic variable 416
SEQ symbolic variable 417
SEQ# symbolic variable 417
SQ symbolic variable 417
SEQ symbolic variable 417
SEQ TEMPLATE descriptor variable 418
DDSEQ symbolic variable 411
RSEQ# symbolic variable 416
SEQ symbolic variable 417
SEQ# symbolic variable 417
SEQ# symbolic variable 417
SEQAUTH installation option 436
SEQDEP installation option 436
SEQUENC installation option 436
sequence object type 185
SESSION command 536
session logs
  actions captured 369
  browsing 369
  enabling 150
  purging 372
session profiles 351
  assigning to users 363
  authorization to create 352
  CLIST 363
  creating initial list filters 359
  CRS option 364
  CUSTOMIZE command 354
  customizing commands table 358
  customizing Primary Menu 354
  deleting 360
  displaying descriptions 357
  displaying list 356
  editing customized commands table 362
  editing customized Primary Menu 361
  PR parameter in ACTEMAIN CLIST 363, 364
  retrieving initial list filters 363
  SET PROFILE command 352
  TAILOR command 358
  types of customization 351
SESSION_LOG table 158
SET command 536
SG command 547
SG object type 39, 179, 182, 185
SHRLEVEL CHANGE 126
Simple Space Estimation (SSE) feature 46, 252
skeleton library compiler. See SLIB (skeleton library) compiler
skeleton library, overriding POF values 139
SLIB (skeleton library) compiler
  changing an ISPF skeleton 385
  compiling 384, 387
  ISPF file tailoring 386
  processing 387
  runtime report 387
  runtime report summary 389
  runtime unit 387
  testing changes 386
SLIB variables, list of 409
SMFPRM member 70
SMS (Storage Management Subsystem) 119
SN symbolic variable 417
SN TEMPLATE descriptor variable 417, 418
solutions, BMC Software 30
SORT command 201, 556, 552
sorting object lists by column 201
SORTPnn data set 108
SORTWK_NBR POF keyword 506
SORTWK_PRIQTY POF keyword 506
SORTWK_SECQTY POF keyword 506
SORTWK_UNIT POF keyword 506
SORTWORK data set 111
SPACE command 542
SPBXPRINT AEXIN keyword 407
specify log option (OPT) 155
SPLAN installation option 432
SPLIT ISPF command 46
SPNAME symbolic variable 417
SQ symbolic variable 417
SQ TEMPLATE descriptor variable 417
SQL
  applying model statements 219
  copying external 218
  copying in SQL_Table 218
  creating new SQL from existing 218
  creating new SQL_Table entry 218
  escape character 82
  extended SQL processing 222
  generating 215
  SQL_Table 217
SQL command 537
SQL command options, setting 89
SQL Explorer for DB2
  ACTPSS CLIST 392
  commands to access SQL 392
  integrating with CATALOG MANAGER 391
  SQLX edit macro 400
SQL SELECT options, setting 92
SQL_Table
  copying SQL 218
  creating new entry 218
  defined 217
  displaying 217
  explaining a statement with Common Explain 398
SQL_TABLE table 158
SQLX_EXP_LOAD POF keyword 506
-SQLM worklist command 531
SQLX edit macro 400
SRTOUT_DATACLASS POF keyword 507
SRTOUT_DATACLASS_ALT POF keyword 507
SRTOUT_EXPDT POF keyword 507
SRTOUT_MGMTCLASS POF keyword 507
SRTOUT_MGMTCLASS_ALT POF keyword 507
SRTOUT_PREFIX POF keyword 507
SRTOUT_STORCLASS POF keyword 507
SRTOUT_STORCLASS_ALT POF keyword 507
SRTOUT_THRESH POF keyword 507
SRTOUT_UNIT POF keyword 507
SRTOUT_UNIT_ALT POF keyword 508
SS command 543
SS symbolic variable 417
SS TEMPLATE descriptor variable 417
SSE command 252
SSE ISPF command 46
SSID
attaching 161
connecting 166
specifying at startup 35
SSID AEXIN keyword 407
SSID OUTPUT descriptor variable
GRPNM symbolic variable 411
JSSID symbolic variable 413
MSSID symbolic variable 415
SS symbolic variable 417
SSID symbolic variable 417
TSSID symbolic variable 418
SSID symbolic variable 417
SSID TEMPLATE descriptor variable
GRPNM symbolic variable 411
JSSID symbolic variable 413
MSSID symbolic variable 415
SSID symbolic variable 417
TSSID symbolic variable 418
SSID symbolic variable 417
SSID symbolc variable 417
STEP command 547
ST object type 39, 186
ST symbolic variable 417
START CLONE command 540
START command 288, 540
START DB2 command 210
STARTOVER AEXIN keyword 406, 407
static SQL 395, 398
STATS AEXIN keyword 407
STATS command 542
STATUS command 542
STEP# symbolic variable 418
STEP.Include_MEMBER POF keyword 508
STEPLIB libraries, setting default values 101
STEPN symbolic variable 417
STEPNAME OUTPUT descriptor variable
STEPN symbolic variable 417
TU1 symbolic variable 418
TU2 symbolic variable 419
TU3 symbolic variable 419
UDOPT symbolic variable 419
WKOWN symbolic variable 420
WKOWNER symbolic variable 420
STEPNAME symbolic variable 417
STEPNAME TEMPLATE descriptor variable
STEPN symbolic variable 417
TU1 symbolic variable 418
TU2 symbolic variable 419
TU3 symbolic variable 419
UDOPT symbolic variable 419
WKOWN symbolic variable 420
WKOWNER symbolic variable 420
STMT installation option 436
STOGROU installation option 436
stogroup object type 39
STOP CLONE command 540
STOP command 286, 540
STOP DB2 command 210
STOPWAIT AEXIN keyword 408
STOPWAIT POF keyword 508
STOPWTSECS AEXIN keyword 408
STOPWTSECS POF keyword 508
storage group object type 185
storage, virtual 70
stored procedure object type 39
stored procedures
ADMIN_COMMAND_DB2 164, 169, 170
ADMIN_DS_LIST 164
DSNWZP 539
DSNWZPT 164
-STOP worklist command 531
STOSPACe command 292, 542
STOSPACE utility worklist command 531
string object type 39, 186
STRINGS installation option 436
SU command 547
SU object type 39, 186
support, customer 3
SUPPRESS_COMMENTS POF keyword 509
SUTnnn data set 108
SWAP ISPF command 46
switches, CATALOG MANAGER 94
switching between BMC Software products 63
SY command 548
SY object type 39, 179, 186
symbolic variables
GDG 116
using in installation options module 409
using on product panels 409
symbolic variables, list of 409
SYNC command 542, 548
SYNDELETE AEXIN keyword 408
SYNDELETE POF keyword 509
SYNLIST AEXIN keyword 408
SYNONYM installation option 436
synonym object type 39, 186
syntax statement conventions 20
SYSCOPY data set, setting JCL options 114
SYSDISC data set 114
SYSELR data set 114
SYSEXEC POF keyword 509
SYSEXIN in BATCH SEARCH 196
SYSSMAP data set 114
SYSLMLIB symbolic variable 418
SYSPROC.ADMN_DS_LIST stored procedure 168
SYSPUNCH data set 114
SYSPREC data sets
LOB 114
ROWID 114
setting JCL options 114
SYSTEM command 537
system privilege user object type 39
system user object type 186
SYSTEM_MLIB POF keyword 509
SYSUID symbolic variable 418
SYST data set 114
SYST_DATACLASS POF keyword 509
SYST_DATACLASS_ALT POF keyword 509
SYST_EXPDT POF keyword 509
SYST_MGMTCLASS POF keyword 509
SYST_MGMTCLASS_ALT POF keyword 510
SYST_PREFIX POF keyword 510
SYST_RETPD POF keyword 510
SYST_SECQTY POF keyword 510
SYST_UNIT ALT POF keyword 511
SYST_UNIT POF keyword 511
TAPE1 POF keyword 511
TAPE2 POF keyword 511
TAPE3 POF keyword 511
tapes
setting default processing options 109
stacking, disabling 111
using 118
TAKEID AEXIN keyword 408
TB command 548
TB object type 39, 179, 182, 186
TBBR command 228
TBCR symbolic variable 418
TBCRE symbolic variable 418
TBEDIT command 235
TBLPROF installation option 436
TBNAM symbolic variable 418
TBNAME symbolic variable 418
TBNODE symbolic variable 418
TBP command 548
TC command 548
tc object type 41, 186
tdsn installation option 432
tdsn option for site profiles 293
technical support 3
TEMPLATE control statements
available IBM utilities 307
creating 306
including in utility jobs 307
TEMPLATE data set 130
TEMPLATE descriptor variables, list of 409
TEMPLATE POF keyword 511
temporary work data sets 112
TEMPUNIT POF keyword 512
TERM command 542
THAW command 552
TB object type 39, 179, 182, 186
TBBR command 228
TBCR symbolic variable 418
TBCRE symbolic variable 418
TBEDIT command 235
TBLPROF installation option 436
TBNAM symbolic variable 418
TBNAME symbolic variable 418
TBNODE symbolic variable 418
TBP command 548
TC command 548
tc object type 41, 186
tdsn installation option 432
tdsn option for site profiles 293
technical support 3
TEMPLATE control statements
available IBM utilities 307
creating 306
including in utility jobs 307
TEMPLATE data set 130
TEMPLATE descriptor variables, list of 409
TEMPLATE POF keyword 511
temporary work data sets 112
TEMPUNIT POF keyword 512
TERM command 542
THAW command 552
threshold, for alternate unit 119
TI symbolic variable 418
TI TEMPLATE descriptor variable 418
TIME OUTPUT descriptor variable
HMS symbolic variable 411
JHMS symbolic variable 412
TI symbolic variable 418
TIME symbolic variable 418
TIME symbolic variable 418
TIME TEMPLATE descriptor variable
HMS symbolic variable 411
JHMS symbolic variable 412
Ti symbolic variable 418
TIME symbolic variable 418
TIME symbolic variable 418
TIME4 symbolic variable 418
TIMEPARM POF keyword 512
TIMESTAMP command 537
TM command 548
TM object type 41
TN command 548
TN object type 179
TNCC installation option 432

T

TA command 353
TBAUTH installation option 436
TABCNST installation option 436
table constraint object type 41, 186
table object type 39, 186
table space mixed object type 41
table space object type 39, 186
table space partition object type 41, 44, 186
table space set object type 41
table space statistics object type 41
table spaces
   dropping 276, 278
   recovering structure and data 283
TABLEPA installation option 436
tables
   CATALOG MANAGER, list of 157
   creating and editing constraints 264
   creating with an existing object 258
   defining columns 260
   for utility support and job generation, list of 157
   identity columns 263
TABLES installation option 436
TABRESH installation option 436
TABLESP installation option 436
TABPRTH installation option 436
TABSTAH installation option 436
TABSTAT installation option 436
TABSTATS command 544
TAILOR command 353, 537
TAPE_EXPDT POF keyword 511
TAPE_RETPD POF keyword 511
TAPE_VOLCNT POF keyword 511

Index 585
TNLMR installation option 432
TP command 548
TP object type 41, 44, 186
TR command 548
TR object type 41, 186
traditional list line format 198, 200
TRIGGER installation option 432, 436
trigger object type 41, 186
troubleshooting
    attaching and connecting to SSIDs 176
    authorization to perform SEARCH 184
    authorization to use logs 367
    availability of actions and object types 361
    customizing Primary Menu 354
    DB2-identifiers 175
drop and drop recovery 278
DROP IS switch 282
DROP RECOVERY function 283
Drop Recovery Logs 374
dropping tables 277
DSNICOPY to recover data 286
generating a list 365
initial list filters 365
objects excluded from recovery 285
pursuing Session Logs 373
recovering changes from logrba 287
recovering incremental image copies 287
recovering indexes 277
referential constraints 277
storage space for dropped tables 277
UNKNOWN COMMAND message 358
WHERE clauses in SEARCH 188, 196
WHERE statements with Session Log Lists 370, 375
wildcard characters in qualifiers 42
TRS installation option 432
TRITCH POF keyword 512
TRUNCATE command 540
truncation, long names 83
trusted context attribute object type 40
trusted context authorization ID object type 40
trusted context object type 40
TS command 548
TS object type 39, 179, 186
TS OUTPUT descriptor variable
    IS symbolic variable 412
    IX symbolic variable 412
    IXNAME symbolic variable 412
    IXSPC symbolic variable 412
    SN symbolic variable 417
    SPNAME symbolic variable 417
    TBNAM symbolic variable 418
    TS symbolic variable 418
    TSIX symbolic variable 418
    TSNAME symbolic variable 418
TS TEMPLATE descriptor variable
    IS symbolic variable 412
    TBNAM symbolic variable 418
    TBNAME symbolic variable 418
    TS symbolic variable 418
    TSNAME symbolic variable 418
TSCR symbolic variable 418
TSIX symbolic variable 418
TSNAME symbolic variable 418
TSO ID (exctsoid) 154
TSO POFRESET command 537
TSO POFRESET POF command 537
TSOPROGRAM POF keyword 512
TSOSUBEXIT POF keyword 512
TSS command 548
TSS object type 41
TSSID symbolic variable 418
TT command 548
TT object type 41, 179
TU1 symbolic variable 418
TU2 symbolic variable 419
TU3 symbolic variable 419
TYPE OUTPUT descriptor variable
    JOBTYP symbolic variable 413
    LOCREM symbolic variable 414
    LR symbolic variable 414
    OBJT symbolic variable 415
    OBJTYP symbolic variable 415
    PB symbolic variable 416
    PRIBAC symbolic variable 416
    RTYPE symbolic variable 416
    RUNTYP symbolic variable 417
    TYPE symbolic variable 419
TYPE parameter 155
TYPE symbolic variable 419
TYPES command 549

U
U line command 553
UA command 548
UA object type 41
Ucmds symbolic variable 419
Ucmds installation option 146, 433
UDOPT symbolic variable 419
UID symbolic variable 419
ULLQ POF keyword 512
ULLQ symbolic variable 419
UN command 548
UN object type 41, 186
UNDO command 552
unit name, ROWID SYSREC data set 118
UNKNOWN COMMAND message 358
UNLD_FREF_DATACLASS POF keyword 513
UNLD_FREF_DIRBLOCK POF keyword 513
UNLD_FREF_MGMTCLASS POF keyword 513
UNLD_FREF_PREFIX POF keyword 513
UNLD_FREF_PRIQTY POF keyword 513
UNLD_FREF SECQTY POF keyword 513
UNLD_FREF_STORCLASS POF keyword 513
UNLD_FREF_SUPPR SUFF POF keyword 513
UNLD_FREF_UNIT POF keyword 514
UNLD1_DATACLASS POF keyword 514
UNLD1_DATACLASS_ALT POF keyword 514
UNLD1_EXPDT POF keyword 514
UNLD1_MGMTCLASS POF keyword 514
UNLD1_MGMTCLASS_ALT POF keyword 514
UNLD1_PREFIX POF keyword 514
UNLD1_PRIQTY POF keyword 514
UNLD1_RETPD POF keyword 515
UNLD1_SECQTY POF keyword 515
UNLD1_STACK POF keyword 515
UNLD1_STORCLASS POF keyword 515
UNLD1_STORCLASS_ALT POF keyword 515
UNLD1_SUPPRESS SUFF POF keyword 515
UNLD1_THRESH POF keyword 515
UNLD1_UNIT POF keyword 515
UNLD1_UNIT ALT POF keyword 516
UNLD2_DATACLASS POF keyword 516
UNLD2_DATACLASS_ALT POF keyword 516
UNLD2_EXPDT POF keyword 516
UNLD2_MGMTCLASS POF keyword 516
UNLD2_MGMTCLASS_ALT POF keyword 516
UNLD2_PREFIX POF keyword 516
UNLD2_PRIQTY POF keyword 516
UNLD2_RETPD POF keyword 517
UNLD2_SECQTY POF keyword 517
UNLD2_STACK POF keyword 517
UNLD2_STORCLASS POF keyword 517
UNLD2_STORCLASS_ALT POF keyword 517
UNLD2_SUPPRESS SUFF POF keyword 517
UNLD2_THRESH POF keyword 517
UNLD2_UNIT POF keyword 517
UNLD2_UNIT ALT POF keyword 518
UNLD3_DATACLASS POF keyword 518
UNLD3_MGMTCLASS POF keyword 518
UNLD3_MGMTCLASS_ALT POF keyword 518
UNLD3_PREFIX POF keyword 518
UNLD3_PRIQTY POF keyword 518
UNLD3_RETPD POF keyword 518
UNLD3_SECQTY POF keyword 518
UNLD3_STACK POF keyword 518
UNLD3_STORCLASS POF keyword 518
UNLD3_STORCLASS_ALT POF keyword 518
UNLD3_SUPPRESS SUFF POF keyword 519
UNLD3_THRESH POF keyword 519
UNLD3_UNIT POF keyword 519
UNLD3_UNIT ALT POF keyword 519
UNLD4_DATACLASS POF keyword 519
UNLD4_MGMTCLASS POF keyword 519
UNLD4_MGMTCLASS_ALT POF keyword 519
UNLD4_PREFIX POF keyword 519
UNLD4_PRIQTY POF keyword 519
UNLD4_RETPD POF keyword 519
UNLD4_STACK POF keyword 519
UNLD4_SUPPRESS SUFF POF keyword 519
UNLD4_UNIT POF keyword 520
UNLOAD command 292, 542
unload data set, used by utilities 107
UNLOAD.+ .LOAD POF keyword 520
UNLOADDOPT AEXIN keyword 408
UNLOADDOPT POF keyword 520
UODSN installation option 433
UP command 552
UP ISPF command 46
UP symbolic variable 419
UPART symbolic variable 419
UPDATE command 540
UPLAN installation option 433
US command 548
US object type 39, 186
user authorization object type 41
user command program 154
user commands table
modifying 146
retaining from previous release 152
user name object type 41, 186
user object type 39, 186
user options
general 71
JCL Generation 72
setting 75
user POF
creating 133
updating directly 134
updating in options panels 134
using multiple 135
user privileges, copying by privilege type 333
user profile data set for user profiles 293
user variables 131
USER_VAR1_CHAR POF keyword 520
USER_VAR2_CHAR POF keyword 520
USER_VAR3_CHAR POF keyword 520
USER_VAR4_CHAR POF keyword 520
USER_VAR5_CHAR POF keyword 520
USER1 symbolic variable 419
USER2 symbolic variable 419
USERAUT installation option 436
USERID OUTPUT descriptor variable
USERID symbolic variable 419
ZACCTNUM symbolic variable 421
ZPREFIX symbolic variable 421
ZUSER symbolic variable 421
USERID symbolic variable 419
USERID symbolics variable 419
USERNAM installation option 436
user-written commands
&CLIST parameter 149
&CMD parameter 150
&DB2MAX parameter 150
&DB2MIN parameter 150
&HELP parameter 150
&LOAD parameter 150
&LOG parameter 150
&LSTO parameter 150
&NLIST parameter 150
&NOSERVER parameter 151
&NUM parameter 151
&OBJECTS parameter 151
&PARSER parameter 151
user-written commands (continued)
&PLAN parameter 151
&WFEK parameter 152
commands table 144
creating 146
object types 156
writing as CLIST 148
writing as program 148
USRCOMND member 146, 153
UT symbolic variable 420
UT TEMPLATE descriptor variable
ALID symbolic variable 409
FCMD symbolic variable 411
JQID symbolic variable 413
UCMD symbolic variable 419
USER1 symbolic variable 419
USER2 symbolic variable 419
UT symbolic variable 420
UTID symbolic variable 420
UTIL symbolic variable 420
UTILID symbolic variable 420
UTILPFX symbolic variable 420
UTILSFX symbolic variable 420
WKID symbolic variable 420
WORKID symbolic variable 420
WORKID8 symbolic variable 421
UTID OUTPUT descriptor variable 420
UTID symbolic variable 420
UTIL command 543
UTIL OUTPUT descriptor variable
ALID symbolic variable 409
FCMD symbolic variable 411
JQID symbolic variable 413
LI symbolic variable 414
LIST symbolic variable 414
OBJT symbolic variable 415
OBJTYP symbolic variable 415
UCMD symbolic variable 419
USER1 symbolic variable 419
USER2 symbolic variable 419
UT symbolic variable 420
UTID symbolic variable 420
UTIL symbolic variable 420
UTILID symbolic variable 420
UTILPFX symbolic variable 420
UTILSFX symbolic variable 420
WKID symbolic variable 420
WORKID symbolic variable 420
WORKID8 symbolic variable 421
UTILITY profile ID command 543
utility processing
BMC Software utilities 290
IBM DB2 utilities 291
JCL data set name 296
JCL options 296
multiple utilities 300
number of control statements allowed 292
single utility 295
User Profile data set name 296
Utility ID variables 297, 298
utility profile data set, creating 293
utility profiles
allocating a data set 293
changing options values 304
creating from an existing profile 302
editing 304
last-used profile ID 305
online tutorial 292
PROFILE command 306
profile ID 299
setting up 293
site profiles 293
User Profile data set name 296
user profiles 293
UTILPFX symbolic variable 420
UTILSFX symbolic variable 420
UVR1 symbolic variable 420
UVR2 symbolic variable 420
UVR3 symbolic variable 420
UVR4 symbolic variable 420
UVR5 symbolic variable 420
V
V line command 553
VCAT symbolic variable 420
version information, displaying 67
view object type 39, 186
VIEW table 158
VIEWDEP installation option 436
VIEWS installation option 433, 437
virtual storage 70
VL command 548
VL object type 41, 186
volume object type 41, 186
VOLUMES installation option 437
VSAM object
data set sizing 106
JCL generation 381
sampling 381
VW command 548
VW object type 39, 179, 182, 186


**W**

Wait-for-Enter commands 59–63
WARNRC AEXIN keyword 409
WDSN installation option 433
WHERE clauses in SEARCH 188
where command is executed parameter (exccsrc) 154
wildcard characters
  - cascading authorizations 325
  - fixed-length CHAR columns 43
  - in host variables 192
  - in quick-search 195
  - in WHERE clause 188
  - two-part object names 44
  - use in Qualifier field 42
  - with saved SEARCH variables 195
WKID symbolic variable 420
WKOWN symbolic variable 420
WKOWNER symbolic variable 420
work data sets
  - permanent 114
  - temporary 112
  - used by utilities 107
WORK_DATACLASS POF keyword 520
WORK_MGMTCLASS POF keyword 520
WORK_STORCLASS POF keyword 521
WORKID AEXIN keyword 409
WORKID symbolic variable 420
WORKID8 symbolic variable 421
worklist commands
  - -BMCU 525
  - -DBUG 526
  - -DSN1 526
  - file format 524
  - -MERG 527
  - -MODI 528
  - -QUI 529
  - -REPO 530
  - -REPX 530
  - -STOS 531
worklist, comment lines 525
WRKnnnn data set 108

**X**

X ALL command 183
X command 537
X line command 183
X line designator 62
XC command 548
XC object type 41
XML relationship object type 41
XML string object type 41
XMLREL installation option 437
XMLSTR installation option 437
Xnnn command 537
XODSN installation option 434
XR command 548
XS command 548
XS object type 41
XSR command 548
XSROBJ installation option 437
XT command 548
XT object type 39, 185
XX command 537

**Y**

YE symbolic variable 421
YE TEMPLATE descriptor variable 421
YEAR OUTPUT descriptor variable 421
YEAR symbolic variable 421
YMD symbolic variable 421
YY OUTPUT descriptor variable 421
YY symbolic variable 421
YYDDD symbolic variable 421
YYYYDDD symbolic variable 421

**Z**

Z line command 553
ZACCTNUM SLIB variable 421
ZACCTNUM symbolic variable 421
ZOOM command 257, 552
ZOOM ISPF command 46
ZPREFIX SLIB variable
  - PREFIX symbolic variable 416
  - ZPREFIX symbolic variable 421
ZPREFIX symbolic variable 421
ZSYSID SLIB variable 421
ZSYSID symbolic variable 421
ZUSER SLIB variable
  - UID symbolic variable 419
  - USERID symbolic variable 419
  - ZUSER symbolic variable 421
ZUSER symbolic variable 421