CATALOG MANAGER for DB2 User Guide

Supporting

Version 11.2.00 of CATALOG MANAGER for DB2
Version 11.2.00 of BMC Object Administration for DB2
Version 11.2.00 of BMC Administrative Assistant for DB2
Version 11.2.00 of System Performance for DB2
Version 11.2.00 of Database Administration for DB2

May 2015
Contacting BMC Software

Several methods are available for 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</td>
<td>1 713 918 8000</td>
</tr>
<tr>
<td>2101 CITYWEST BLVD</td>
<td></td>
<td>or</td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827 USA</td>
<td></td>
<td>1 800 841 2031</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>


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.

Explorer is a trademark or registered trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

DB2, DB2 Connect, DB2 Universal Database, FlashCopy, IBM, IBM z, OS/390, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Oracle and Java are registered trademarks 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

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 problems similar to yours and possible solutions
■ Order or download product documentation
■ Download products and maintenance
■ Report a problem or ask a question
■ Subscribe to receive proactive e-mail alerts
■ 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 problem
■ 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
  — Messages from related software
License key and password information

If you have questions about your license key or password, contact Customer Support through one of the following methods:

- Send an e-mail message to customer_support@bmc.com. (In the Subject line, enter SupID:yourSupportContractID, such as SupID:12345.)
- In the United States and Canada, call 1 800 537 1813. Outside the United States and Canada, contact your local support center for assistance.
Contents

About this book .................................................................15
Related publications ........................................................................................................15
Conventions .....................................................................................................................16
Syntax statements ...........................................................................................................16

Chapter 1 Overview of CATALOG MANAGER .........................................................19
CATALOG MANAGER features .................................................................................19
List of DB2 catalog objects .........................................................................................20
Object creation .............................................................................................................20
DB2 catalog search .......................................................................................................21
Object description .........................................................................................................21
Data manipulation .........................................................................................................21
Command and statement execution ............................................................................21
Utility profiles ..............................................................................................................22
Dropped object recovery ..............................................................................................22
Action log maintenance ..............................................................................................22
Remote DB2 subsystem access .....................................................................................23
Update and execution of SQL statements ...................................................................23
Authorization management .........................................................................................24
Indirect catalogs ..........................................................................................................24
SQL Explorer execution ...............................................................................................25
Support for DB2 Analytics Accelerator for z/OS ......................................................25
Integration with BMC solutions ..................................................................................26
Administrative Assistant for DB2 ..............................................................................26
Database Administration for DB2 ..............................................................................26
System Performance for DB2 ......................................................................................27
BMC Object Administration .......................................................................................27
Where to go from here .................................................................................................27

Chapter 2 Getting started with CATALOG MANAGER ..........................................29
Accessing CATALOG MANAGER functions .............................................................29
Using the Primary Menu panel ...............................................................................31
Using the Command line ..........................................................................................32
Using a command recognition character ..................................................................33
Working in pop-up windows .......................................................................................33
Selecting an Action .....................................................................................................34
| Setting data set options | .......................................................... | 84 |
| Setting SQL and confirm options | .......................................................... | 86 |
| Setting SQL SELECT options | .......................................................... | 88 |
| Setting panel graphic options | .......................................................... | 89 |
| Setting CATALOG MANAGER switches | .......................................................... | 90 |
| Setting DESCRIBE options | .......................................................... | 93 |
| Setting product options | .......................................................... | 94 |
| Setting the JCL options for job cards | .......................................................... | 96 |
| Setting the JCL options for STEPLIBs | .......................................................... | 98 |
| Setting the JCL options for static data sets | .......................................................... | 100 |
| Setting the JCL options for tapes | .......................................................... | 104 |
| Setting the JCL options for temporary work data sets | .......................................................... | 107 |
| Setting the JCL options for permanent data sets | .......................................................... | 109 |
| Setting the JCL generation data group options | .......................................................... | 115 |
| Setting the JCL debugging display and execution options | .......................................................... | 117 |
| Setting the JCL utility installation options module name options | .......................................................... | 119 |
| Setting the online reorg options | .......................................................... | 120 |
| Setting the non-worklist JCL options | .......................................................... | 122 |
| Setting the LISTDEF and TEMPLATE data set options | .......................................................... | 126 |
| Setting user variables | .......................................................... | 127 |
| Creating a user POF | .......................................................... | 127 |
| Updating a user POF | .......................................................... | 129 |
| Using multiple POFS | .......................................................... | 130 |
| Refreshing the initial POF | .......................................................... | 131 |
| Generating POF reports | .......................................................... | 132 |
| Reusing a POF in a subsequent installation | .......................................................... | 133 |
| Overriding POF values in SLIBs | .......................................................... | 134 |
| Adding steps to the JCL | .......................................................... | 135 |
| Obtaining a list of TEMPLATEs or LISTDEFS in CATALOG MANAGER | .......................................................... | 138 |
| Using the commands table | .......................................................... | 138 |
| Writing user commands as CLISTs | .......................................................... | 146 |
| Development aids for user commands | .......................................................... | 147 |
| Command program parameters | .......................................................... | 147 |
| Passing object type and name | .......................................................... | 149 |
| CATALOG MANAGER tables | .......................................................... | 152 |
| Where to go from here | .......................................................... | 153 |

**Chapter 4**  
**Accessing other DB2 subsystems**

| Using the DB2 Attach feature | .......................................................... | 155 |
| Attaching CATALOG MANAGER to a specified SSID | .......................................................... | 156 |
Attaching to an SSID or server by using the connection selection list ........................................157
Switching catalog access ........................................................................................................158
Restoring the default attachment .........................................................................................159
Using the DB2 Connect feature ............................................................................................160
Connecting to a specified SSID ............................................................................................161
Obtaining data set information from a remote SSID ..........................................................164
Connecting to a remote SSID from a location list ...............................................................165
Connecting to a remote SSID and issuing DB2 commands to manipulate objects ............165
Using saved connections ......................................................................................................166
Viewing the connections table .............................................................................................168
Using DB2-identifiers with the CONNECT command ..........................................................170
Identification of attachments or connections with unique values ....................................170
Displaying the current server ID ..........................................................................................171
Troubleshooting a DB2 Attach or DB2 Connect failure .......................................................172
Where to go from here ............................................................................................................172

Chapter 5        Working with lists and searches 173
Using mixed lists ..................................................................................................................173
Valid source objects for mixed lists ..................................................................................174
Generating a mixed list .......................................................................................................174
Combining lists ....................................................................................................................176
Generating a combined list ..................................................................................................176
Excluding objects from a combined list ..............................................................................177
Using SEARCH to generate lists based on object attributes ..............................................178
Valid objects for searches ....................................................................................................178
Generating a list by using the SEARCH command ............................................................180
Using host variables in a search .........................................................................................185
Using the Quick-Search feature ..........................................................................................187
Using saved search variables in a Quick-Search ...............................................................188
Using a WHERE clause in a Quick-Search .........................................................................188
Creating complex searches ...............................................................................................189
Creating searches that do not contain a JOIN ....................................................................189
Creating searches that contain a JOIN ................................................................................190
Customizing object list displays .......................................................................................191
Specifying a new order for displayed columns ....................................................................191
Using the traditional list line format ..................................................................................192
Sorting a list by one or more columns ...............................................................................193
String value search .............................................................................................................194
Counting items ...................................................................................................................195
Describing list objects .........................................................................................................196
DESCRIBE command ................................................................. 197
DES command ...................................................................... 198
D and S commands ............................................................... 199
DESTATISTICS command .................................................. 199
Printing lists ....................................................................... 200
Generating JCL for a job in batch ................................................. 201
Using the BATCH command for a DB2 object list or a mixed list .............................................................................. 201
Using the BATCH command for a CATALOG MANAGER list or search ................................................................. 205
Generating editing and executing SQL ........................................... 206
Confirm SQL panels ................................................................ 206
Using the SQL_Table ................................................................ 207
Applying SQL model statements .............................................. 209
Using extended SQL processing ............................................. 212
Where to go from here ........................................................ 214

Chapter 6 Browsing and editing data ........................................... 215

Browsing table data ............................................................... 215
Methods for invoking the data browsing function ...................... 216
Setting options for browsing data ......................................... 216
Browsing data ....................................................................... 221
Browsing data in LOB columns ............................................. 222

Editing table data ................................................................. 222
Methods for invoking the data editing function ....................... 223
Setting options for editing data ............................................ 223
How CATALOG MANAGER handles lock contention .................. 226
Editing data ......................................................................... 226

Copying table data ............................................................... 230
Using the COPY command .................................................. 230
Using the Copy Table Rows option ........................................ 233

Where to go from here ........................................................ 236

Chapter 7 Creating objects .......................................................... 239

Before you begin ................................................................... 239
Space requirement estimates based on user-specified values ............... 240
Estimating space requirements for a table space ....................... 240
Estimating space requirements for an index ............................... 243
Using an existing object as a model to create objects ..................... 244
Considerations for creating objects ........................................ 244
Example of creating a table ................................................... 245
Generating DDL to create objects .......................................... 255
Appendix A  JCL Generation 355
JCL Generation data sets sizing function ........................................... 356
Example of Worklist JCL ........................................................................... 357

Appendix B  Using the Skeleton Library compiler 359
SLIB compilation ............................................................................................ 361
SLIB changes ........................................................................................................ 361
   SLIB verification using ISPF file tailoring .............................................. 362
   Compilation of changed SLIBs .................................................................. 363
SLIB processing ................................................................................................. 363
   Generating the SLIB report ....................................................................... 364

Appendix C  Integrating CATALOG MANAGER with the Common Explain component 367
Before you begin .............................................................................................. 367
Commands to access SQL .................................................................................. 367
   Explaining a DBRM package or plan ....................................................... 369
   Explaining a statement from a DBRM or package ................................ 370
   Explaining a statement from the SQL_Table .......................................... 373
   Editing the SQL statement by using the SQLX edit macro ...................... 374

Appendix D  JCL Generation keywords and variables 377
AEXIN keywords ............................................................................................ 377
Symbolic variables for BMC Administrative products .................................. 386

Appendix E  CATALOG MANAGER installation options 401
Installation option example ............................................................................. 401
Installation option descriptions ....................................................................... 404

Appendix F  JCL Generation product options 415
Example of product options ........................................................................... 415
Descriptions of product option keywords ..................................................... 423

Appendix G  CATALOG MANAGER worklist commands 493
Worklist file format .......................................................................................... 493
Worklist commands .......................................................................................... 495
   -BMCU (Execute a BMC utility) ............................................................... 495
   -DBUG (Debug) ...................................................................................... 495
   -DSN1 (IBM DSN1COPY utility) ......................................................... 496
   -MERG (IBM MERGECOPY utility) ..................................................... 496

Where to go from here .......................................................................................... 353
-MODI (IBM MODIFY utility) ................................................................. 497
-NOOP (No operation) ................................................................. 497
-QUI (IBM QUIESCE utility) ............................................................. 498
-REPO (IBM REPORT utility) ............................................................ 498
-REPX (IBM REPAIR utility) ............................................................. 499
-SQLM (Group multiple ALTER statements) ..................................... 499
-STOS (IBM STOSPACE utility) ....................................................... 499

Appendix H Commands 501

CATALOG MANAGER commands ........................................................... 501
DB2 action commands ................................................................................ 506
Utility commands ........................................................................................ 508
Utility list commands .................................................................................. 512
Statistics commands .................................................................................... 513
List commands .......................................................................................... 513
User commands .......................................................................................... 519
Data browsing and editing commands ....................................................... 519
  Command line commands ........................................................................ 519
  Line commands ......................................................................................... 523

Glossary 525
About this book

This book contains detailed product information and is intended for application developers and database administrators (DBAs).

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).

Related publications

From the BMC Support Central website, you can use the following methods to access related publications that support your product or solution:


- View Quick Course videos (short overviews of selected product concepts, tasks, or features), which are available from the following locations:
  - Documentation Center
  - Support Central (at http://www.bmc.com/support/mainframe-demonstrations)
  - BMC Mainframe YouTube channel (https://www.youtube.com/user/BMCSoftwareMainframe)

You can order hardcopy documentation from your BMC sales representative or from the support site. You can also subscribe to proactive alerts to receive e-mail messages when notices are issued.

Tip
You can access the BMC Support Central site at http://www.bmc.com/support.

Conventions

This document uses the following special conventions:

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

- Variable text in path names, system messages, or syntax is displayed in italic text: testsys/instance/fileNome

- Menu sequences use a symbol to convey the sequence. For example, Actions => Create Test instructs you to choose the Create Test command from the Actions menu.

Syntax statements

This topic explains conventions for showing syntax statements.

A sample statement follows:

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

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

The CATALOG MANAGER for DB2 product provides an efficient gateway to DB2 catalog information. 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).

CATALOG MANAGER features

CATALOG MANAGER includes highly productive features for creating and managing your DB2 databases:

- List of DB2 catalog objects
- Object creation
- DB2 catalog search
- Object description
- Data manipulation
List of 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.

Object creation

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.
DB2 catalog search

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.

For more information, view the Quick Course "CATALOG MANAGER for DB2 - New and improved Search support."

Object description

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.

Data manipulation

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.

Command and statement execution

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
CATALOG MANAGER features

- BMCSTATS and 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.

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

Dropped object recovery

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.

Action log maintenance

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.

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

Remote DB2 subsystem access

CATALOG MANAGER provides the following methods of accessing remote 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.

Update and execution of SQL statements

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.

Authorization management

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 the list of user or objects, 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.

Indirect catalogs

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

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

The SQL Performance 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

Support for DB2 Analytics Accelerator for z/OS

CATALOG MANAGER supports DB2 Analytics Accelerator for z/OS.

CATALOG MANAGER lets you select a table and perform the following operations:

- Add table to an accelerator
- Modify distribution or organizing keys
- Move data from DB2 to an accelerator
- Return the current definition and status information of an accelerator
- Copy data from DB2 to an accelerator
- Remove data and a table from an accelerator
- Restore data that was moved to an accelerator by an ARCHIVE TABLES to DB2
- Enable or disable use of a loaded table on the accelerator
- Enable or disable incremental updates for a table

For more information, view the Quick Course "CATALOG MANAGER for DB2 - Support for DB2 Analytics Accelerator for z/O."
Integration with BMC solutions

CATALOG MANAGER is a component of the following BMC solutions:

- BMC Administrative Assistant for DB2
- Database Administration for DB2
- BMC Object 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 these documents:

- BMC Administrative Assistant for DB2 release notes
- Installation System Reference Manual
- Installation System Quick Start
- BMC Products and Solutions for DB2 Customization 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 these documents:

- Database Administration for DB2 release notes
- Installation System Reference Manual
- Installation System Quick Start
- BMC Products and Solutions for DB2 Customization 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 these documents:

- BMC System Performance for DB2 release notes
- BMC System Performance for DB2 User Guide
- Installation System Reference Manual
- Installation System Quick Start
- BMC Products and Solutions for DB2 Customization Guide

BMC Object Administration

You can use the BMC Object Administration for DB2 solution to manage your DB2 databases quickly, efficiently, and effectively.

For more information, see these documents:

- BMC Object Administration for DB2 release notes
- Installation System Reference Manual
- Installation System Quick Start
- BMC Products and Solutions for DB2 Customization 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.

“Getting started with CATALOG MANAGER” on page 29 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

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 Reference Manual, Installation System Quick Start, and the BMC Products and Solutions for DB2 Customization Guide.

In addition:

- In your ISPF settings, remove the / for the Long message in pop-up option.
- Configure a TSO region size of 8 MB or greater for each user who signs on.

For more information, view the Quick Course "Getting Started."

Accessing CATALOG MANAGER functions

The BMC Software Administrative products for DB2 panel is provided to make it easy to start CATALOG MANAGER and interact with the other ISPF-based Administrative products.
Note
Product selections that are displayed on the panel depend on the products that you have installed.

Figure 1: Example of BMC Software Administrative Products for DB2 panel

```
COMMAND ===> _________________________________________________________________
   _ 1 DASD MANAGER for DB2      - Manage or monitor DB2 physical objects
   2 CATALOG MANAGER for DB2    - Execute DDL or query DB2 Catalog
   3 CHANGE MANAGER for DB2    - Manage changes to DB2 objects/structures

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

To start CATALOG MANAGER from the BMC Software Administrative Products for DB2 panel

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
   You cannot specify the same SSID as 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 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 attach to an installed copy of the specified subsystem catalog.

4. In the Use Shared or Individual product ISPF APPLID field, specify one of the following types of ISPF Application IDs (APPLID):
   - To specify an APPLID for each DB2 subsystem, type **S** (shared).
To specify an APPLID for each product, type I (individual).

5 Press Enter.

The CATALOG MANAGER Primary Menu panel is displayed.

**Figure 2: CATALOG MANAGER Primary Menu panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0. (L)</td>
<td>List using customizable lists</td>
</tr>
<tr>
<td></td>
<td>1. (S)</td>
<td>Search for catalog objects</td>
</tr>
<tr>
<td></td>
<td>2. (C)</td>
<td>Create objects</td>
</tr>
<tr>
<td></td>
<td>3. (G)</td>
<td>Grant privileges</td>
</tr>
<tr>
<td></td>
<td>4. (O)</td>
<td>CATALOG MANAGER options processing</td>
</tr>
<tr>
<td></td>
<td>5. (D)</td>
<td>DB2 Commands</td>
</tr>
<tr>
<td></td>
<td>6. (M)</td>
<td>Maintain logs menu</td>
</tr>
<tr>
<td></td>
<td>7. (Q)</td>
<td>List SQL for edit &amp; execution</td>
</tr>
<tr>
<td></td>
<td>8. (R)</td>
<td>About this Release/CATALOG MANAGER Quick Reference</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12. SG Stogroup</td>
<td>17. SU SysPrivUser</td>
<td>22. AL Alias</td>
<td>27. LO Location</td>
</tr>
<tr>
<td></td>
<td>13. TB Table</td>
<td>18. SY Synonym</td>
<td>23. US User</td>
<td>28. CK Checks</td>
</tr>
<tr>
<td></td>
<td>15. IX Index</td>
<td>20. CI Collection</td>
<td>25. DM DBRM</td>
<td>30. XT Aux Tabl</td>
</tr>
</tbody>
</table>

Qualifier %.%

Initial attach to DEEG

Copyrights (c) 1987-2015 BMC Software, Inc.
as an unpublished licensed work.
All rights reserved.

The CATALOG MANAGER 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 on page 32.
Table 1: Description of Primary Menu panel

<table>
<thead>
<tr>
<th>Panel element</th>
<th>Description or use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title line</td>
<td>This line includes the following information:</td>
</tr>
<tr>
<td></td>
<td>■ SSID to which you are attached</td>
</tr>
<tr>
<td></td>
<td>■ Product name and version number</td>
</tr>
<tr>
<td></td>
<td>■ Panel title</td>
</tr>
<tr>
<td>Command line</td>
<td>The user enters CATALOG MANAGER commands, TSO commands, and ISPF commands</td>
</tr>
<tr>
<td>Action field</td>
<td>The user enters the corresponding number or letter of an action from the list of actions</td>
</tr>
<tr>
<td>Obj type field</td>
<td>The user enters the corresponding number or code of the DB2 object with which to work</td>
</tr>
<tr>
<td>Qualifier field</td>
<td>The user enters a character string that identifies the objects with which to work</td>
</tr>
</tbody>
</table>

Using the Command line

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

To issue a command

1. 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 34.

   ■ 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 “JCL Generation product options” on page 415 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 CATALOG MANAGER processes it as such.

**To force ISPF to ignore a command**

1. Type the > command recognition character before the command with no intervening spaces (for example, >PRINT).

### Working in pop-up windows

CATALOG MANAGER provides pop-up windows for many fields that can accept specific discrete values. Pop-up windows enable you to enter values by selecting a value, instead of typing it.

Pop-up windows are available throughout CATALOG MANAGER. They are not necessarily indicated in the user interface, they must be discovered through use.

**To work in a pop-up window**

1. Next to a suitable field in the user interface, enter a question mark (?) instead of a value.

2. On the pop-up screen that is displayed, navigate the cursor to the required value.

3. Click **Enter**.

The field is populated with the selected value.

In the following example, the user has entered ? in the **Locksize** field, the **Locksize** pop-up window is displayed. Only the pop-up window is available, the remainder of the screen is inactive.
**Note**

The value of the field will depend on the position of the cursor when the user clicks **Enter**, not the position of select (S), which is not required in the pop-up window. If the user moves the cursor to ROW, then clicks **Enter**, the value of the **Locksize** field will be ROW not ANY (where S has been entered).

---

**Selecting an Action**

Each **Action** on the Primary Menu panel invokes a CATALOG MANAGER command.

**To choose an Action**

1. Use one of the following methods:
   - On the **Command** line, type the corresponding number.
   - In the **Action** field, type the corresponding number.
   - In the **Action** field, type the corresponding letter (shown in parentheses) that labels the desired action.
Note

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

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

For more information, see “Identifying an object type” on page 35.

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.

“Accessing CATALOG MANAGER functions” on page 29 and Table 2 on page 35 show the DB2 object types that are displayed on the Primary Menu panel, and their object type codes.

To choose an object type

1. Type the appropriate two-letter code to choose an object type.

   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>
</tbody>
</table>
## Object type code

<table>
<thead>
<tr>
<th>Object type code</th>
<th>Object type</th>
</tr>
</thead>
<tbody>
<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 1.a</td>
<td>Location</td>
</tr>
<tr>
<td>CK</td>
<td>Check constraint</td>
</tr>
<tr>
<td>PR</td>
<td>Stored procedure</td>
</tr>
<tr>
<td>XT</td>
<td>Auxiliary table 1.b</td>
</tr>
</tbody>
</table>

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

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

---

Table 3 on page 36 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

<table>
<thead>
<tr>
<th>Object type code</th>
<th>Object type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AU</td>
<td>Authorization 1.a</td>
</tr>
<tr>
<td>BP</td>
<td>Buffer pool</td>
</tr>
<tr>
<td>CA</td>
<td>Column authorization</td>
</tr>
<tr>
<td>CD</td>
<td>Check dependent</td>
</tr>
<tr>
<td>CL</td>
<td>Column label 1.a</td>
</tr>
<tr>
<td>Object type code</td>
<td>Object type</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CP</td>
<td>Constraint dependent</td>
</tr>
<tr>
<td>CX</td>
<td>Trusted context</td>
</tr>
<tr>
<td>CXA</td>
<td>Trusted context authorization ID&lt;sup&gt;1.1&lt;/sup&gt;</td>
</tr>
<tr>
<td>CXT</td>
<td>Trusted context attribute&lt;sup&gt;1.1&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>Dependency</td>
</tr>
<tr>
<td>DS</td>
<td>Data set&lt;sup&gt;1.1&lt;/sup&gt;</td>
</tr>
<tr>
<td>DT</td>
<td>Data or distinct type</td>
</tr>
<tr>
<td>EN</td>
<td>Environment variables</td>
</tr>
<tr>
<td>FK</td>
<td>Foreign key&lt;sup&gt;1.1&lt;/sup&gt;</td>
</tr>
<tr>
<td>FN</td>
<td>Function routine</td>
</tr>
<tr>
<td>FO</td>
<td>Routine option</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;1.1&lt;/sup&gt;</td>
</tr>
<tr>
<td>IL</td>
<td>IP list</td>
</tr>
<tr>
<td>IM</td>
<td>Index mixed&lt;sup&gt;1.1&lt;/sup&gt;</td>
</tr>
<tr>
<td>IN</td>
<td>IP name&lt;sup&gt;1.2&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;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>IXC</td>
<td>Index cleanup</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>Java path</td>
</tr>
<tr>
<td>KC</td>
<td>Key column&lt;sup&gt;1.1&lt;/sup&gt;</td>
</tr>
<tr>
<td>KT</td>
<td>Key-target</td>
</tr>
<tr>
<td>KTD</td>
<td>Key-target distribution&lt;sup&gt;1.2&lt;/sup&gt;</td>
</tr>
<tr>
<td>Object type code</td>
<td>Object type</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>KTH</td>
<td>Key-target history</td>
</tr>
<tr>
<td>KTS</td>
<td>Key-target statistics</td>
</tr>
<tr>
<td>KU</td>
<td>Key column user</td>
</tr>
<tr>
<td>LK</td>
<td>Limit key</td>
</tr>
<tr>
<td>LL</td>
<td>LU list</td>
</tr>
<tr>
<td>LM</td>
<td>LU mode</td>
</tr>
<tr>
<td>LO</td>
<td>Accessible remote locations (DDF)</td>
</tr>
<tr>
<td>LS</td>
<td>LU mode select</td>
</tr>
<tr>
<td>LU</td>
<td>LU names</td>
</tr>
<tr>
<td>MQT</td>
<td>Materialized query table</td>
</tr>
<tr>
<td>MQ</td>
<td>MQ (main menu)</td>
</tr>
<tr>
<td>MX</td>
<td>Mixed object types</td>
</tr>
<tr>
<td>NP</td>
<td>Native SQL procedure</td>
</tr>
<tr>
<td>OB</td>
<td>Online schema changes</td>
</tr>
<tr>
<td>OS</td>
<td>LOBSTATS</td>
</tr>
<tr>
<td>PA</td>
<td>Plan authorization</td>
</tr>
<tr>
<td>PI</td>
<td>Packlist</td>
</tr>
<tr>
<td>PK</td>
<td>Primary key</td>
</tr>
<tr>
<td>PM</td>
<td>Row permissions</td>
</tr>
<tr>
<td>PT</td>
<td>Index and table space partition</td>
</tr>
<tr>
<td>RD</td>
<td>Object role dependency</td>
</tr>
<tr>
<td>RE</td>
<td>Relation</td>
</tr>
<tr>
<td>RI</td>
<td>Referential integrity</td>
</tr>
<tr>
<td>RO</td>
<td>Roles</td>
</tr>
<tr>
<td>SC</td>
<td>Schema</td>
</tr>
<tr>
<td>SE</td>
<td>Identity column</td>
</tr>
<tr>
<td>SU</td>
<td>System privileges for AUTHIDS</td>
</tr>
<tr>
<td>TC</td>
<td>Table constraint</td>
</tr>
<tr>
<td>TM</td>
<td>Table space mixed</td>
</tr>
</tbody>
</table>
### Table 4: Four-character object types

<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>

**Table 4 on page 39** lists the four-character object type codes that are available from the **Command** line of a KT list.
Specifying object qualifiers

You can use object qualifiers to further describe the object type that you want to list.

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.

**To use an object qualifier**

1. Specify an object qualifier in the **Qualifier** field of the CATALOG MANAGER Primary Menu panel (see “Accessing CATALOG MANAGER functions” on page 29).

   **Tip**

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

**Supported wildcards in qualifiers**

To generate a list of objects that match more than one character string, you can include one of the following wildcard characters in the qualifier.

Table 5 on page 40 describes the wildcard characters that CATALOG MANAGER supports.

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

**Note:** If the _ Wild switch is set to N, an underscore will not be considered as a wildcard character if no other wildcard characters (such as % or *) are included in the qualifier for a table list. For more information about switches, see “Setting CATALOG MANAGER switches” on page 90.
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 on page 41 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 on page 41, 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>

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 the following table 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>Complete two-part object name</td>
</tr>
<tr>
<td>Is followed by a period</td>
<td>First part of the two-part object name</td>
</tr>
<tr>
<td>Does not include a period</td>
<td>Second part of the two-part object name</td>
</tr>
</tbody>
</table>

**Qualifier exceptions**

Certain qualifiers have additional uses and requirements.

Table 8 on page 42 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.DTBNAMEN.</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 X'F200F1'.

Table 9 on page 42 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>
</tbody>
</table>
### ISPF commands

Most of the ISPF commands in the product operate in the same manner as they do in other ISPF applications.

Table 10 on page 43 describes the most commonly used ISPF commands.

#### Table 10: ISPF commands

<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&lt;br&gt;&lt;<strong>More:</strong> + on a panel indicates that more information is available below the current line.&lt;br&gt;Scrolling is available on a Model 2 3270 mainframe terminal, which uses a 24-line by 80-column display.</td>
</tr>
<tr>
<td>END (or F3)</td>
<td>Validates and processes information, the same as the <strong>Enter</strong> key&lt;br&gt;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&lt;br&gt;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&lt;br&gt;&lt;<strong>More:</strong> &lt; on a panel indicates that more information is available to the left.&lt;br&gt;On the CATALOG ROW panel, when you press F10, the previous object in the object list is displayed.</td>
</tr>
<tr>
<td>PFSHOW</td>
<td>Displays the active function keys&lt;br&gt;Some panels use every available line to display input variables. To display all variables, enter <strong>PFSHOW OFF</strong> on the <strong>Command</strong> line.</td>
</tr>
</tbody>
</table>
### Command Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETURN (or (\text{=X}))</td>
<td>Returns to the Primary Menu panel</td>
</tr>
<tr>
<td>RIGHT (or (\text{F11}))</td>
<td>Scrolls the panel to the right&lt;br&gt;<strong>More: &gt;</strong> on a panel indicates that more information is available to the right.&lt;br&gt;On the CATALOG ROW panel, when you press F11, the next object in the object list is displayed.</td>
</tr>
<tr>
<td>SPLIT (or (\text{F2}))</td>
<td>Divides the panel and displays the ISPF Primary Option Menu in the new panel&lt;br&gt;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 (\text{F9}))</td>
<td>Switches from one split panel to another</td>
</tr>
<tr>
<td>UP (or (\text{F7}))</td>
<td>Scrolls the panel up&lt;br&gt;<strong>More: -</strong> on a panel indicates that more information is available above the current line.&lt;br&gt;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 (\text{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&lt;br&gt;In an object list, you must type \text{S} in the \text{Cmd} column to display the full value of an object with a long name.&lt;br&gt;The data editing function does not support the (\text{F4}) key.</td>
</tr>
</tbody>
</table>

## Getting help

CATALOG MANAGER provides you with panel-level Help.

**To obtain Help for a panel**

1. 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.
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 on page 45 shows some of the more common paths through the CATALOG MANAGER panels.

Figure 3: Navigating through CATALOG MANAGER

Generating lists in CATALOG MANAGER

Using the LIST function (the most fundamental feature of CATALOG MANAGER), you can generate lists of objects from the DB2 catalog.

You can then easily manipulate the listed objects by issuing CATALOG MANAGER and DB2 commands against the objects.
List panels

This topic shows a sample list panel and briefly describes list panel features.

**Figure 4: Object list**

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Database Owner</th>
<th>Stogroup</th>
<th>Buf Pool</th>
<th>DBID</th>
<th>ROShr</th>
<th>Type</th>
<th>Group</th>
<th>Encode</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUDAC</td>
<td>ASUQA</td>
<td>QZUALL</td>
<td>BP0</td>
<td>1622</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUDA1</td>
<td>ASUQA</td>
<td>QZUALL</td>
<td>BP0</td>
<td>402</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUB81</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1346</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUB82</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1343</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUB83</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1347</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUB84</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1348</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUB85</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1350</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUB86</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1351</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUCCF</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>615</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUC115</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP8K0</td>
<td>1352</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUC119</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1377</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUC101</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1378</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUC102</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1355</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUC103</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1356</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QZUC104</td>
<td>ASUQA</td>
<td>SYSDEFLT</td>
<td>BP0</td>
<td>1359</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following features are included:

- The panel's 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
A stack level indicator below the scroll increment indicator

In Figure 4 on page 46, 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 48.

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 76.

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 (see Figure 5 on page 48).

2 Press Enter.
CATALOG MANAGER displays a secondary list of the table spaces in the source database (see Figure 6 on page 49).

**Figure 6: Table Space List panel**

![Table Space List panel](image)

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** (see Figure 7 on page 49).

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

![Generating table lists from multiple table spaces](image)

2. Press Enter.
CATALOG MANAGER first displays a secondary list of tables within the first source table space (see Figure 8 on page 50).

**Figure 8: Table List panel for first source table space**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMD</td>
<td>CSR</td>
</tr>
</tbody>
</table>

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

Lists: ACCTB AL CA CD CK CL CO CP C2 DB DP DS DT FK IC IM IS IX KC KU LK MK
QUALIFIER: TABLESPACE=QZUDA1.QZUS01A1

<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>QZU.QZUT01_DA1S01</td>
<td>QZUDA1</td>
<td>21</td>
<td>T</td>
<td>2036</td>
<td>128</td>
</tr>
</tbody>
</table>

---

3 Press END.

CATALOG MANAGER displays the list of tables in the next source table space (see Figure 9 on page 50).

**Figure 9: Table List panel for next source table space**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMD</td>
<td>CSR</td>
</tr>
</tbody>
</table>

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

Lists: ACCTB AL CA CD CK CL CO CP C2 DB DP DS DT FK IC IM IS IX KC KU LK MK
QUALIFIER: TABLESPACE=QZUDA1.QZUS02A1

<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>QZU.QZUT01_DA1S02</td>
<td>QZUDA1</td>
<td>4</td>
<td>T</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QZU.QZUT02_DA1S02</td>
<td>QZUDA1</td>
<td>0</td>
<td>T</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---

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 on page 50) 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 on page 51).
Press Enter.

**Figure 10: Generating lists of varied dependent objects**

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: ACCTB AL CA CD CK CL CO CP C2 DB DP DS DT FK IC IM IS IX KC KU LK MK
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--------
IX QZU.QZUT01_DA1S02 QZUDA1 QZUS02A1 21 1 T 2036 128
KC QZU.QZUT02_DA1S02 QZUDA1 QZUS02A1 4 0 T 0 0
******************************  BOTTOM OF DATA  ******************************

CATALOG MANAGER first displays a secondary list of the indexes in the first source table (see Figure 11 on page 51).

**Figure 11: Index List panel for first source table**

DEFF-R --------------------------  INDEX LIST  -------------------- ROW 1 OF 5
Command ===>                                                  Scroll ===> CSR 04
CMD will show commands for this list. Type command and press ENTER
Lists: BMCUHIST CO DB DS IC IM IP IS ISS IX KC KT KTD KTDH KTDS KTH KTS LK NP
QUALIFIER: TABLE=QZU.QZUT01_DA1S02
C Index Name                  Table Name                  UT Cl  Col Keys
-------v--------1--------v--------2--------v--------3--------v--------4--------v--------5--------v--------6--------v--------7--------v--------
QZU.QZUX01_DA1S02T01 QZU.QZUT01_DA1S02 P2 NY 1 2036
QZU.QZUX02_DA1S02T01 QZU.QZUT01_DA1S02 D2 NN 6 2036
QZU.QZUX03_DA1S02T01 QZU.QZUT01_DA1S02 D2 NN 19 2036
QZU.QZUX04_DA1S02T01 QZU.QZUT01_DA1S02 D2 YN 19 2036
QZU.QZUX05_DA1S02 QZU.QZUT01_DA1S02 U2 NN 19 2036
******************************  BOTTOM OF DATA  ******************************

2 Press END.

CATALOG MANAGER displays a secondary list of the key columns in the second source table (see Figure 12 on page 51).

**Figure 12: Column List panel for second source table**

DEFF-R ------------------------  KEY COLUMN LIST  ----------------- ROW 1 OF 2
Command ===>                                                  Scroll ===> CSR 04
CMD will show commands for this list. Type command and press ENTER
Lists: CL CO KC
QUALIFIER: TABLE=QZU.QZUT02_DA1S02
CmIndex Name                      Column               Seq Num O   Lth DatTyp
-------v--------1--------v--------2--------v--------3--------v--------4--------v--------5--------v--------6--------v--------7--------v--------
QZU.QZUX01_DA1S02T02 EMPNO                  1 2 A 9 CHAR
QZU.QZUX01_DA1S02T02 SSSNO                  2 1 A 9 CHAR
******************************  BOTTOM OF DATA  ******************************

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.

Related Information

- “Commands” on page 501

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

Perform the following task to display 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 on page 53).

Figure 13: Scrollable Commands List panel

<table>
<thead>
<tr>
<th>DEGA-R</th>
<th>Commands List</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>==&gt;</td>
<td>CSR</td>
</tr>
</tbody>
</table>

Enter S to see the HELP panel for the command.

<table>
<thead>
<tr>
<th>Command</th>
<th>Type</th>
<th>Valid List Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLY</td>
<td>Cat P</td>
<td>*Valid on cmd line of any list</td>
</tr>
<tr>
<td>AUDIT</td>
<td>Cat P</td>
<td>*Valid on cmd line of any list</td>
</tr>
<tr>
<td>BATCH</td>
<td>Cat P</td>
<td>*Valid on cmd line of any list</td>
</tr>
<tr>
<td>BROWSE</td>
<td>Cat P</td>
<td>MQT TB VW</td>
</tr>
<tr>
<td>CANCEL</td>
<td>Cat P</td>
<td>*Valid on cmd line of any list</td>
</tr>
<tr>
<td>CASCADE</td>
<td>Cat P</td>
<td>SU US</td>
</tr>
<tr>
<td>CATALOGHELP</td>
<td>Cat P</td>
<td>*Valid on cmd line of any list</td>
</tr>
<tr>
<td>CLIPBOARD</td>
<td>Cat P</td>
<td>*Valid on cmd line of any list</td>
</tr>
<tr>
<td>CLIST</td>
<td>Cat P</td>
<td>*Valid on cmd line of any list</td>
</tr>
<tr>
<td>CMD</td>
<td>Cat P</td>
<td>*Valid on cmd line of any list</td>
</tr>
<tr>
<td>COMMANDS</td>
<td>Cat P</td>
<td>*Valid on cmd line of any list</td>
</tr>
<tr>
<td>COPYAUTHS</td>
<td>Cat P</td>
<td>BP CI DB DT FN MOT NP PG PL PR SC SE SG SU TB TS UA US VW</td>
</tr>
<tr>
<td>CUSTOMIZE</td>
<td>Cat P</td>
<td>*Valid on cmd line of any list</td>
</tr>
</tbody>
</table>

Accessing information about a specific command

You can display the syntax and description for a specific command from the Commands List panel, from the Primary Menu panel, or from an object list panel.

To display information from the Commands List panel

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

To display information from the Primary Menu panel or an object list panel

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

You can list all of the valid commands for an object list.

To display a list of commands

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.

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.

Abbreviated 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.

Omitted object types in 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.
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:

- “To issue a command against individually selected objects” on page 56
To issue a command against individually selected objects

1. In the **Cmd (C)** column beside each source object name, type the command (Figure 14 on page 56).

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

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

   ```
   DEFF-R ------------------------  TABLESPACE LIST  ----------------- ROW 1 OF 5
   Command ===>                                                  Scroll ==> CSR
   02
   CMD will show commands for this list. Type command and press ENTER
   Lists: ACCTB AL BMCUHIST CA CL CO DB DS FK IC IM IS IX LK MQT MX NP OS PA PDD
   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----
   BMCCOPY1.QZUS01A1   ASUQA      4 BP0       0    1       130        A    A
   QZUDA1.QZUS02A1   ASUQA     16 BP0       0    2       146        A    A
   BMCCOPY1.QZUS03A1   ASUQA      0 BP0       4    1      1440        A    A
   QZUDA1.QZUS04A1   ASUQA     64 BP0       0    2       540        A    A
   BMCCOPY1.QZUS05A1   ASUQA      0 BP0       4    1       720        A    A
   ******************************  BOTTOM OF DATA  ******************************
   ```

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

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

![Figure 15: Tablespace List panel with shortcut commands for copying multiple table spaces](image)

3 Press Enter.

To issue a command against all objects in the list

1 Type the command followed by the keyword ALL on the Command line of the list panel.

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

![Figure 16: Tablespace List panel with command to copy all listed table spaces](image)

To exclude listed objects

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: Tablespace List panel with exclude commands**

![Figure 17: Tablespace List panel with exclude commands](image)
2 Press Enter to process the exclusion.

CATALOG MANAGER displays the panel with the marked objects.

3 On the Command line, type the appropriate command followed by the keyword ALL.

CATALOG MANAGER processes the command against all objects except those objects excluded by the X designator.

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

1 On the Command line of the current product, enter the command corresponding to the product to which you want to switch (see Table 11 on page 58).

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>BMCCCAT</td>
</tr>
<tr>
<td>DASD MANAGER PLUS for DB2</td>
<td>BMCDASD</td>
</tr>
</tbody>
</table>

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

For more information about enabling Fast Path Navigation, see:

- Installation System Reference Manual
- Installation System Quick Start
- BMC Products and Solutions for DB2 Customization Guide
Where to go from here

In this section, you learned the basics of how to use CATALOG MANAGER to manage your DB2 catalog.

“Setting up CATALOG MANAGER” on page 61 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 section 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

You can view information about the version of CATALOG MANAGER that you are running.

To view information

1. From the Primary Menu panel or any list panel, type the ENVIRONMENT (ENVI) command on the Command line.

   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

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.
To view DB2 initialization parameters

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

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

You can display the CATALOG MANAGER common area in dump format. The common area stores many of the internal values for the CATALOG MANAGER installation.

To view the CATALOG MANAGER common area

1 From the Primary Menu panel or any list panel, type **PEEK** on the **Command** line.

*Note*

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 view and update DB2 special registers

1 From the Primary Menu panel or any object list panel, type the **SEE** command on the **Command** line.

**Figure 18**: DB2 Special Registers panel

```plaintext
DEEG-R  DB2 Special Registers  1 to 30 of 30
Command ==>  Scroll ==>

Catalog Manager Nickname . . . .
Current Values
User . . . . . . . . . . . . . MVSJXE1
Current SQLID . . . . . . . MVSJXE1
Application Encoding Scheme . 37
Current Date . . . . . . . 07/29/2014
Current Degree . . . . . . . 1
Local LC_CTYPE . . . . . . .
Table Types for Optimization . SYSTEM
Data Sharing Member Name . .
Current Optimization Hint . .
Package Path . . . . . . . .
Current Package Set . . . .
Current Path . . . . . . . . "SYSIBM","SYSPFUN","SYSPROC","MVSJXE1"
Current Precision . . . . . DEC15
Refresh Age . . . . . . . . 0
Current Rules . . . . . . . DB2
Schema . . . . . . . . . . . MVSJXE1
Current Server . . . . . . . DEEG
Current Time . . . . . . . 08.35.45
Current Timestamp . . . . . 2014-07-29-08.35.45.745528
Current Time Zone . . . . . GMT -5.00
Current Debug Mode . . . . . DISALLOW
Current DECFL0AT Rounding Mode . ROUND_HALF_EVEN
Current Routine Version . .
Client Accounting . . . . .
Client Application Name . . MVSJXE1
Client User ID . . . . . . . MVSJXE1
Client Workstation Name . . DB2CALL
```

**Note**

When the value for the **Current Package Set** field is blank, all packages in the pack list are available to the user.

2 Update the values of any or all of the following fields on the DB2 Special Registers panel by typing over the existing value:

- **Current Application Compatability**
- **Current SQLID**
- **Current Degree**
- **Local LC_CTYPE**
- **Table Types for Optimization**
- **Current Optimization Hint**
- **Current Path**
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 on page 64 lists the plans that CATALOG MANAGER provides. In addition to the product code for CATALOG MANAGER (ACT), the plan names include the version number (v) and the release number (r).

Table 12: CATALOG MANAGER plans

<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>Plan name</td>
<td>Function name</td>
<td>Plan description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>------------------</td>
</tr>
<tr>
<td>ACTvrDE</td>
<td>Data Editing and Browsing</td>
<td>Enables access to data editing and browsing functions. 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. ACT vrDH 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. 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. ACT vrDL 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. This plan allows the minimum access that is required to use CATALOG MANAGER. The ACT vrDM 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>
</tbody>
</table>
### Plan manipulation commands

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 “JCL Generation product options” on page 415.

### Setting the MEMLIMIT system parameter

CATALOG MANAGER requires above-the-bar memory and might abend if sufficient memory is not available.

1. Use any of the following methods to override the default value of the MEMLIMIT value:

   - Specify the MEMLIMIT parameter in the JCL
   - Specify REGION=0M in the JCL
   - Use the SMF IEFUSI exit
Note
The default value for the System Management Facility (SMF) MEMLIMIT parameter is 2 GB. This value is set in member SMFPRMxx in SYS1.PARMLIB.

Using options to control your environment

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

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

These options provide you with the ability to:

- Tailor the interface
- Set up defaults for generating job statements
- Specify defaults for parameters, names, and prefixes for allocated data sets

Typically, the person who installs CATALOG MANAGER sets default values for user options. For information about establishing installation option values at installation, see:

- BMC Products and Solutions for DB2 Customization Guide
- Installation System Quick Start
- Installation System Reference Manual

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 module is ACTDOPD1. The source of the installation option modules is located in the HLQ.UBMCCNTL 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.

**To display the installation options**

1. 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.

**Related Information**

- “CATALOG MANAGER installation options” on page 401

**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.
Storing values in the ISPF profile

The ISPF profile resides in the data set member prdxPROF, where prdx is the value of the application ID in the BMCDB2 CLIST. To define or modify the values in this profile, you can use the Options panels of ALTER or CHANGE MANAGER, JCL Generation, or CM/PILOT. You can use literal characters or symbolic variables to specify the values on the Options panels. (For more information about symbolic variables, see the ALTER and CHANGE MANAGER for DB2 Reference Manual.)

1. To save your user options in your profile, exit the product.

   **Note**
   If your ISPF session abnormally ends (abends), the user options that you modified are not saved.

To refresh an option value in all existing ISPF profiles

You can refresh the values in your user options by editing and reassembling the installation options module.

1. 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 refreshing user options, see:

- *Installation System Reference Manual*
- *Installation System Quick Start*
- *BMC Products and Solutions for DB2 Customization 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.UBMCCNTL 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 uses 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 CATALOG MANAGER product. Thus, when you specify an option for generating JCL in one product, your selection applies to all of the products.

**Note**
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 BMC1010.ACM.D10.LOAD and then you installed version 11.2.00 of the product. Update the value of the keyword to a version 11.2.00 LINK library (BMC1120.ACM.D11.BMCLINK).

- 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.
How CATALOG MANAGER uses options

The following figure 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.

**To set user options**

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**.

**Note**

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.

**To access the options panels**

CATALOG MANAGER provides two fast path methods for accessing the options panels.

1. Perform one of the following tasks:
   
   - From the **Command** line of any panel, type **OPT number**, where number is the number assigned to an options panel (see Table 13 on page 73), and press **Enter**. The options panel is displayed.

**Tip**

Pressing **END** returns to the previous panel.

---

**Table 13: Options panel**

<table>
<thead>
<tr>
<th>Number</th>
<th>Panel name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Options</td>
</tr>
<tr>
<td>Number</td>
<td>Panel name</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------</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>Data sets</td>
</tr>
<tr>
<td>5</td>
<td>SQL and Confirm Options</td>
</tr>
<tr>
<td>6</td>
<td>SQL Select</td>
</tr>
<tr>
<td>7</td>
<td>Colors</td>
</tr>
<tr>
<td>8</td>
<td>Switches</td>
</tr>
<tr>
<td>9</td>
<td>Describe Options</td>
</tr>
</tbody>
</table>

- From the **Command** line of the Options panel, type `number`, where `number` is the number assigned to an options panel (Table 13 on page 73), and press **Enter**. The options panel is displayed.

2. To return to the Options panel, press **END**.

**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.

**To define an options data set**

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: Options Dataset not defined panel**

```
------------------------- Options Dataset not defined -----------------------
Command ===>                                                        
CATALOG MANAGER now requires a dataset to store your user option settings.
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.
```

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 on page 76). Go to Step 4 on page 76.

**Figure 21: Allocate Data Set**

<table>
<thead>
<tr>
<th>Command</th>
<th>Allocate data set with the following values</th>
<th>N (Y/N)</th>
</tr>
</thead>
</table>

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

| Volume Serial | (Blank for authorized default volume) | * |
| Generic Unit | SYSALLDA | (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.

### Setting basic options

The fields on the Options panel 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.

**To set the options values**

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.

**Figure 22: CATALOG MANAGER Options panel**

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Options</th>
<th>Scroll ===&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>User options Dsn . . . . .</td>
<td>RDACRJ.BMCCAT.USEROPT(ACTUSR)</td>
<td></td>
</tr>
<tr>
<td>Describe pds member . . . .</td>
<td>RDACRJ</td>
<td></td>
</tr>
<tr>
<td>DB2 VCAT . . . . . . . .</td>
<td>DEFFCAT</td>
<td></td>
</tr>
<tr>
<td>Default SQLID . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max lines per list . . . . .</td>
<td>0</td>
<td>0-9999, 0-Unlimited</td>
</tr>
<tr>
<td>Maximum # of select lines .</td>
<td>300</td>
<td>0-9999, 0-Unlimited</td>
</tr>
<tr>
<td>Profile . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1) Edit General options . .</td>
<td>N</td>
<td>Y/N Edit General options</td>
</tr>
<tr>
<td>2) Edit Object selections . .</td>
<td>N</td>
<td>Y/N Edit Mixed list and HDDL options</td>
</tr>
<tr>
<td>3) Edit JCL Generation options</td>
<td>N</td>
<td>Y/N Edit JCLGEN and PDF options</td>
</tr>
<tr>
<td>4) Edit Dataset names . . .</td>
<td>N</td>
<td>Y/N Edit Dataset names</td>
</tr>
<tr>
<td>5) Edit SQL and Confirm options</td>
<td>N</td>
<td>Y/N Edit SQL and Confirm panel options</td>
</tr>
<tr>
<td>6) Edit SQL Select options .</td>
<td>N</td>
<td>Y/N Edit SQL Select options</td>
</tr>
<tr>
<td>7) Edit Color options . . .</td>
<td>N</td>
<td>Y/N Edit Color settings</td>
</tr>
<tr>
<td>8) Edit Switches . . . .</td>
<td>N</td>
<td>Y/N Edit Switches</td>
</tr>
<tr>
<td>9) Edit Describe options . .</td>
<td>N</td>
<td>Y/N Edit User Describe options</td>
</tr>
</tbody>
</table>

3 (optional) In the **User options Dsn** field, you can 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: Options Dataset has changed panel**

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your CATALOG MANAGER Options data set has changed.</td>
</tr>
<tr>
<td>Options dataset: RDACRJ.BMCCAT.USEROPT(ACTUSR)</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>Action . . . : N (Y/N)</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

- 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**.
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.

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

---

**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 93.

---

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 on page 78 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 <strong>Default SQLID</strong> field.</td>
</tr>
<tr>
<td>*PERSIST</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>The SQL ID that was current when you ended your last CATALOG MANAGER session can be any of the following values:</td>
</tr>
<tr>
<td></td>
<td>■ An SQL ID that was set in the <strong>Default SQLID</strong> field</td>
</tr>
<tr>
<td></td>
<td>■ The last SQL ID that was set by using the SET sql ID command during the previous session</td>
</tr>
<tr>
<td></td>
<td>■ An SQL ID that was set by previous use of the *PERSIST value in the <strong>Default SQLID</strong> field</td>
</tr>
</tbody>
</table>

---

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 “Customizing CATALOG MANAGER command access” on page 329.

10 Press END to return to the Primary Menu panel.

### Setting general options

You can change the general options settings.

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

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: General Options panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>General Options</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUBDYN1</td>
<td>AUBDYN1</td>
<td>PAGE</td>
</tr>
<tr>
<td>Decimal point . . .</td>
<td>Period or comma</td>
<td>Scroll</td>
</tr>
<tr>
<td>SQL string delimiter</td>
<td>Quote or double-quote</td>
<td>Scroll</td>
</tr>
<tr>
<td>Lines per page</td>
<td>75</td>
<td>0-199, 0 - unlimited</td>
</tr>
<tr>
<td>DB2 feedback size</td>
<td>64</td>
<td>16-999, Size of feedback area in KB</td>
</tr>
<tr>
<td>Truncation chars</td>
<td>..</td>
<td>Long name truncation character</td>
</tr>
<tr>
<td>Truncation position</td>
<td>L</td>
<td>L-Left, M-Middle, R-Right</td>
</tr>
<tr>
<td>Char field max width</td>
<td>30</td>
<td>10-99 For list display</td>
</tr>
<tr>
<td>Terse level</td>
<td>VERBOSE</td>
<td>TERSE/VERBOSE</td>
</tr>
</tbody>
</table>

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 191.
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.

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

## Setting object use options

You can set the 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: Object Use Options panel

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

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

   Table 15 on page 82 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 commands produce.

**Note**

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

The HDDL command uses a temp data set to build the resulting DDL. If you want to execute the HDDL line command, you must edit the Product Options File (POF) or create a new POF, as the values required to allocate the data set cannot be specified through the CATALOG MANAGER interface.

The temp DATASET is allocated based on values in the POF data set for the following variables.

- ACTWRK_DATACLASS =
- ACTWRK_MGMTCLASS =
- ACTWRK_PRIQTY = 10 (recommended default value)
- ACTWRK_SECQTY = 2 (recommended default value)
- ACTWRK_STORCLASS =
- ACTWRK_UNIT = SYSALLDA (recommended default value)

The device where the data set is saved is defined according to the first POF variable, in the following order, that contains a valid value:

1. ACTWRK_MGMTCLASS
2. ACTWRK_STORCLASS
3. ACTWRK_DATACLASS
4. ACTWRK_UNIT

For details, see “Creating a user POF” on page 127 or “Updating a user POF” on page 129.
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.

**To set JCL Generation 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 JCL Generation options** field, type **Y**.

   The JCL Generation Options panel is displayed.

   **Figure 26: JCL Generation Options panel**

   ![](image)

<table>
<thead>
<tr>
<th>Command ===</th>
<th>JCL Generation Options 1 to 6 of 6</th>
<th>Scroll ===&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separate steps</td>
<td>N</td>
<td>Y/N Separate job step for each statement</td>
</tr>
<tr>
<td>Generate using Worklist.</td>
<td>N</td>
<td>Y/E/N Yes/Yes with Events/No</td>
</tr>
<tr>
<td>Object limit</td>
<td>300</td>
<td>0-9999, 0 - unlimited</td>
</tr>
<tr>
<td>Group attach</td>
<td></td>
<td>Group attach for Data Sharing</td>
</tr>
<tr>
<td>Edit JCLgen options</td>
<td>N</td>
<td>Y/N Edit PDF options</td>
</tr>
</tbody>
</table>

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.
To generate the utility JCL by

<table>
<thead>
<tr>
<th>Type</th>
<th>Using a worklist</th>
<th>Not using a worklist</th>
<th>Using a worklist and to include event information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>E</td>
</tr>
</tbody>
</table>

If you have the DASD MANAGER PLUS product installed, DASD MANAGER PLUS inserts rows into the DASD MANAGER PLUS EVENTS table.

The choices that you make for the **Separate steps** and the **Generate using Worklist** fields determine how utility statements are created:

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

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

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

8 In the **Edit JCLGen options** field, type Y to display a series of panels for setting options for the product options file. For more information, see “**Setting product options**” on page 94.

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

**Setting data set options**

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

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

To set data set options

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

2. Press Enter.

   The Options panel is displayed.

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

   The Datasets panel is displayed.

   Figure 27: Datasets panel

   ![Datasets panel]

   1 to 11 of 11

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 “Creating a user utility profile data set” on page 279.

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

10 Press END to return to the Options panel.

Setting SQL and confirm options

On the SQL and Confirm Options panel, you can specify the default settings of options that are used by SQL commands, as well as the options that are displayed on Confirm SQL panels.

Confirm SQL panels are found near the end of many CATALOG MANAGER processes. They display a summary of the commands to be written to the SQL statement that will process the desired actions. Set the values that you expect to use most frequently for all SQL processing to avoid having to set them every time that you execute an SQL command.

Note

CATALOG MANAGER uses a 2-MB work area for processing SQL.

To set SQL 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 SQL and Confirm options field, type Y.

The SQL and Confirm Options panel is displayed.

**Figure 28: SQL and Confirm Options panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>SQL and Confirm Options</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL processing</td>
<td>INSERT,DELETE,UPDATE,CREATE and executed from the SQL Table</td>
<td>PAGE</td>
</tr>
<tr>
<td>Auto-commit</td>
<td>Y</td>
<td>Y/N</td>
</tr>
<tr>
<td>Generate SQL/DSN</td>
<td>N</td>
<td>Y/N</td>
</tr>
<tr>
<td>Confirm SQL and DSN panel defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit</td>
<td>N</td>
<td>Y/N</td>
</tr>
<tr>
<td>Save in SQL table</td>
<td>N</td>
<td>A/Y, A/Y-Append, R-Replace, N-No</td>
</tr>
<tr>
<td>Execute</td>
<td>N</td>
<td>Y/N</td>
</tr>
<tr>
<td>Last used is default</td>
<td>N</td>
<td>Y/N</td>
</tr>
<tr>
<td>Drop recovery and revoke reassign defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add dependency list</td>
<td>N</td>
<td>Y/N</td>
</tr>
<tr>
<td>Drop recovery on</td>
<td>N</td>
<td>Y/N</td>
</tr>
<tr>
<td>Log image copies</td>
<td>N</td>
<td>Y/N</td>
</tr>
<tr>
<td>Add cascade report</td>
<td>N</td>
<td>Y/N</td>
</tr>
<tr>
<td>Plan table</td>
<td>PLAN_TABLE</td>
<td></td>
</tr>
</tbody>
</table>

4 Specify your default options for processing SQL:

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

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

b In the **Generate SQL/DSN** field, type Y or N to specify whether the default value for the Generate SQL field is Y or N on the Create or Generate panels.

5 Specify your default options for the confirmation panels:

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

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

<table>
<thead>
<tr>
<th>To default to this action</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Append the SQL to the SQL in the SQL_Table</td>
<td>A</td>
</tr>
<tr>
<td>Save the SQL in the SQL_Table</td>
<td>Y</td>
</tr>
<tr>
<td>Replace the SQL in the SQL_Table</td>
<td>R</td>
</tr>
<tr>
<td>Discard the SQL</td>
<td>N</td>
</tr>
</tbody>
</table>
c In the **Execute** field, type **Y** or **N** to specify the default value for the **Execute SQL** field.

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

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

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

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

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

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

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

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

---

### Setting SQL SELECT options

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

**To set SQL SELECT 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 SQL Select options** field, type **Y**.
The SQL Select panel is displayed.

**Figure 29: SQL Select panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>SQL Select</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse SQL output dataset . . . Y</td>
<td>Y/N Y-Automatically browse dataset</td>
<td></td>
</tr>
<tr>
<td>Maximum output line length . . . 256</td>
<td>1-4092</td>
<td></td>
</tr>
<tr>
<td>Maximum numeric field width . . . 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum char field width . . . 64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum varchar field width . . . 64</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

**To set panel graphic 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 Color options** field, type **Y**.

The Colors panel is displayed.

**Figure 30: Colors panel**

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Colors</th>
<th>1 to 11 of 11</th>
<th>Scroll ===&gt; PAGE</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>

---

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.

#### To set the CATALOG MANAGER switches

1 Perform one of the following tasks:

   - From the **Command** line of the Primary Menu panel or an object list panel, issue one of the following commands:

     — **SET switch ON**

     — **SET switch OFF**

Setting the switch is only temporary for the session. When you start a new CATALOG MANAGER session, the switch is reset.
From the Switches panel, type over the existing value with Y or N.

To access the Switches panel, follow the steps in “To set switches” on page 91.

**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: Switches panel**

<table>
<thead>
<tr>
<th>Command ===</th>
<th>Switches</th>
<th>1 to 6 of 6</th>
<th>Scroll ===</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<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 SOLID before GRANT</td>
<td>N</td>
<td>Y/N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dev debug . . .</td>
<td>. . .</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. To change the value of a switch, type Y or N over the existing value.

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

5 In the **Define No** field, type **Y**, **N**, or **C** to indicate how to include the DEFINE parameter in DDL or HDDL for a table space or index.

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

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

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

7 Press **END** to return to the Options panel.
Setting DESCRIBE options

The DESCRIBE command displays detailed information that is stored in the DB2 catalog about a specific object, including the object’s structure and dependencies. The DESCRIBE command options control how this detailed information is presented for plans, packages, tables, and views.

For more information about the DESCRIBE command, see “Describing list objects” on page 196.

To set DESCRIBE options

Note
To modify the Describe options, in the Basic Options panel, you must specify a valid member name (other than NONE) in the Describe pds member field. For more details, see “Setting basic options” on page 76.

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: Describe Options panel

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

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

5. In the Output dsn field, type the name of the data set in which CATALOG MANAGER writes the DESCRIBE report.
6 Press **END** to return to the Options panel.

## Setting product options

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.

### To set product options

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

2. Press **Enter**.

   The Options panel is displayed.

3. In the **Edit JCL Generation options** field, type **Y** and press **Enter**.

   The JCL Generation Options panel is displayed.

4. In the **Edit JCLgen options** field, type **Y** and press **Enter**.

   The JCL Generation Update - Main Menu panel is displayed.

**Figure 33: JCL Generation Update Panel**

```
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 96</td>
</tr>
<tr>
<td>Steplib Options</td>
<td>“Setting the JCL options for STEPLIBs” on page 98</td>
</tr>
<tr>
<td>Static Data Set Options</td>
<td>“Setting the JCL options for static data sets” on page 100</td>
</tr>
<tr>
<td>Tape Options</td>
<td>“Setting the JCL options for tapes” on page 104</td>
</tr>
<tr>
<td>Individual Data Set Options</td>
<td>■ “Setting the JCL options for temporary work data sets” on page 107</td>
</tr>
<tr>
<td></td>
<td>■ “Setting the JCL options for permanent data sets” on page 109*</td>
</tr>
<tr>
<td>Generation Data Group Options</td>
<td>“Setting the JCL generation data group options” on page 115</td>
</tr>
<tr>
<td>(GDGs)</td>
<td></td>
</tr>
<tr>
<td>Debugging, Display and</td>
<td>“Setting the JCL debugging display and execution options” on page 117</td>
</tr>
<tr>
<td>Execution Options</td>
<td></td>
</tr>
<tr>
<td>Utility Options</td>
<td>“Setting the JCL utility installation options module name options” on page 119</td>
</tr>
<tr>
<td>Listdef and Template Data Sets</td>
<td>“Setting the LISTDEF and TEMPLATE data set options” on page 126</td>
</tr>
<tr>
<td>User Defined Variable Values</td>
<td>“Setting user variables” on page 127</td>
</tr>
<tr>
<td>Product Options File (POF)</td>
<td>■ “Creating a user POF” on page 127</td>
</tr>
<tr>
<td>Functions</td>
<td>■ “Updating a user POF” on page 129</td>
</tr>
<tr>
<td></td>
<td>■ “Using multiple POFs” on page 130</td>
</tr>
<tr>
<td></td>
<td>■ “Refreshing the initial POF” on page 131</td>
</tr>
<tr>
<td></td>
<td>■ “Generating POF reports” on page 132</td>
</tr>
<tr>
<td></td>
<td>■ “Reusing a POF in a subsequent installation” on page 133</td>
</tr>
<tr>
<td></td>
<td>■ “Overriding POF values in SLIBs” on page 134</td>
</tr>
<tr>
<td></td>
<td>■ “Adding steps to the JCL” on page 135</td>
</tr>
<tr>
<td></td>
<td>■ “Obtaining a list of TEMPLATEs or LISTDEFs in CATALOG MANAGER” on page 138</td>
</tr>
</tbody>
</table>

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 on page 96):

<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 <strong>options processing</strong></td>
</tr>
<tr>
<td>Options</td>
<td><strong>Y</strong> at Edit JCL Generation options</td>
</tr>
<tr>
<td>JCL Generation Options</td>
<td><strong>Y</strong> 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**

   ![Figure 34: JCL Generation Jobcard Options Update panel](image)

   Type data and press Enter.

   Is a TSO submit exit used to generate jobcards? . . N (Y/N)

   Enter Jobcards below:
   
   ```
   //MVSCAL2I JOB (&ZACCTNUM),+'&PGMR', // CLASS=A,MSGLEVEL=(1,1),NOTIFY=&SYSUID
   /*
   /*
   Jcllib . . . . . . \n   Sysexec. . . . . . \n   Region size . . . OM \n   Memlimit. . . . . NOLIMIT \n   Time parameter . . . \n   System MLIB. . . . SYS1.0000.ISPMENU \n   Runtime HLQ. . . . ADM.INSTXXXX \n   User HLQ . . . . . USER.HLQ \n   LLQ . . . . . . . . \n   ULLQ . . . . . . . . (Leave blank if 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 **Jclib** 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 **User HLQ** field, type the high-level qualifier used for the user-defined data sets for the installation environment.
**Note**
The User HLQ field is used only if the LLQ field is blank.

12 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.

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

**Note**
If the User HLQ field contains a value, the ULLQ value is ignored.

14 Press END to save your changes and return to the JCL Generation Update - Main Menu panel ALTER or CHANGE MANAGER Main Menu.

### Setting the JCL options for STEPLIBs

Use the Options panels of the JCL Generation component to define or modify the values in your ISPF profile and a user POF.

Use the STEPLIB Options panel to specify the load libraries that appear in jobs that ALTER or CHANGE MANAGER creates.

#### To set the JCL options

1. Use the following menu selections to display the JCL Generation STEPLIB Options Update panel (Figure 35 on page 99):

<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>
</tbody>
</table>
From this menu | Select this item and press Enter
---|---
JCL Generation Update - Main Menu | Steplib Options

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

```
COMMAND ===> Type data and press Enter.
DSNEXIT . . . . . . . SYS3.&SSID..DSNEXIT
DB2 DSNLOAD . . . . . . CSGI.DB2V&DB2V2.M.DSNLOAD
Override lib . . . . . .
CATALOG MANAGER . .
ALTER/CHANGE MANAGER.
DASD MANAGER PLUS .
EXECUTION . . . . . .
COPY PLUS . . . . . .
REORG PLUS . . . . . .
LOADPLUS . . . . . .
UNLOAD PLUS . . . .
RECOVER PLUS . . . .
CHECK PLUS . . . . .
SQL EXPLORER.
Additional lib . . .
IDA LOAD 1 . . . . .
IDA LOAD 2 . . . .
```

2 Specify the data set names for the libraries:

*Tip*

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

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

*Tip*

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

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

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

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

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

3 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel Main Menu.
### 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: JCL Generation Static Data Set Options panel](image)

   **Figure 36: JCL Generation Static Data Set Options panel**

   ```
   COMMAND ===> JCL GENERATION STATIC DATA SET OPTIONS UPDATE
   Type data and press Enter.
   Data set sizing option... C  (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... 200  (Cylinders, 1 - 99999)
   Max secondary quantity.. 20  (Cylinders, 1 - 99999)
   Max unit count................. (Blank or 1 - 59 volumes)
   Include data set cleanup step... Y  (Y/N)
   Return code for cleanup step... 0  (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 on page 101.

   **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 “JCL Generation data sets sizing function” on page 356.
### Table 16: Data set sizing options

<table>
<thead>
<tr>
<th>For this method</th>
<th>Type</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>No data set sizing</td>
<td>N</td>
<td>The product uses the default primary and secondary quantities for the permanent data sets. To modify the quantities for the primary and secondary space for a permanent data set, see Step 4 on page 101.</td>
</tr>
</tbody>
</table>
| Sizing by using statistics from the DASD MANAGER PLUS tables | B    | - The DASD MANAGER PLUS product must be installed and interacting with ALTER or CHANGE MANAGER to use this option.  
- If a column is defined as a LONG VARCHAR, this option averages the row size.  
- Current BMCSTATS statistics should be available for objects in the worklist before you select this option. |
| Sizing by using statistics from the DB2 catalog           | C    | - If a column is defined as a LONG VARCHAR, this option does not average the row size.  
- Current IBM RUNSTATS catalog statistics should be available for objects in the worklist before you select this option. |
| Estimating sizes based on physical, random sampling of VSAM data sets | O    | - If a column is defined as a LONG VARCHAR, this option averages the row size.  
- You can use this option if the statistical information in the DB2 catalog or in the DASD MANAGER PLUS tables is not current. However, JCL generation might take additional time to complete. |

You can specify whether to include comments in the Execution JCL that show statistics for determining data set sizes. For more information, see “Setting the JCL debugging display and execution options” on page 117.

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:

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

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

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

If you want to use multiple data sets on DASD, specify a value greater than 1 for **Max unit count**. On the JCL Generation Individual Data Set Options Update panels (see “Setting the JCL options for permanent data sets” on page 109), 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 on page 102 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)</td>
<td>SYSDnnnn</td>
<td>LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Error</td>
<td>SYSERnnn</td>
<td>CHECK PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM CHECK DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Map</td>
<td>SYSMAP</td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Punch</td>
<td>SYSPUNCH</td>
<td>REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM REORG</td>
</tr>
</tbody>
</table>

---

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

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

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

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

If you want to use multiple data sets on DASD, specify a value greater than 1 for **Max unit count**. On the JCL Generation Individual Data Set Options Update panels (see “Setting the JCL options for permanent data sets” on page 109), 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 on page 102 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)</td>
<td>SYSDnnnn</td>
<td>LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Error</td>
<td>SYSERnnn</td>
<td>CHECK PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOADPLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM CHECK DATA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Map</td>
<td>SYSMAP</td>
<td>IBM LOAD</td>
</tr>
<tr>
<td>Punch</td>
<td>SYSPUNCH</td>
<td>REORG PLUS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IBM REORG</td>
</tr>
</tbody>
</table>
Work data set | ddname | Used in JCL cleanup by |
--- | --- | --- |
Unload (SYSREC) | SYSRnnnn or Rnnnnyyy | REORG PLUS  
IBM REORG  
The unload data sets that are used by REORG PLUS and IBM REORG are deleted automatically. |
Work | Not applicable | Utilities that are listed in Table 18 on page 103. |

Table 18 on page 103 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>
</table>
| SORTOUT  
SORTPnnn  
SORTOnnn | CHECK PLUS  
LOADPLUS  
REORG PLUS  
IBM CHECK DATA  
IBM LOAD  
IBM REORG |
| SYSUTnnn  
SUTnnn  
WRKnnn | CHECK PLUS  
LOADPLUS  
REORG PLUS  
RECOVER PLUS  
IBM CHECK DATA  
IBM LOAD  
IBM REORG  
IBM RECOVER INDEX  
IBM REBUILD INDEX |

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.

**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 Main Menu.

### 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 on page 105):

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

**Figure 37: JCL Generation Tape Options Update panel**

```
COMMAND ===> 
Type data and press Enter.
Tape Unit 1 . . . . . . . . . CART (blank or TAPE, CART, etc)
Tape Unit 2 . . . . . . . . . TAPE (blank or TAPE, CART, etc)
Tape Unit 3 . . . . . . . . . TAPE (blank or TAPE, CART, etc)
Tape Volume count . . . . . 99 (0 - 255)
Tape EXPDT. . . . . . . . . . (Blank or YYDDD or YYYY/DDD)
Tape RETPD. . . . . . . . . . (Blank or 0 - 9999 days)
Tape TRTCH. . . . . . . . . . (Blank or C.E.T,ET,COMP,NOCOMP)
Stacking Options
Local Primary Copy . . . N (Y/N) Local Backup Copy . . . N (Y/N)
Recovery Primary Copy . N (Y/N) Recovery Backup Copy . . N (Y/N)
Primary Sysrec. . . . . N (Y/N) Backup Sysrec . . . . . N (Y/N)
Baseline Recovery . . . N (Y/N) Archive . . . . . . . . N (Y/N)
(CHANGE MANAGER only)                                   |
```

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

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

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

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

6. In the **Tape TRTCH** field, type the parity, data conversion, translation, and compression value for 7-track tape drives as shown in Table 19 on page 105.

**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>
</tbody>
</table>
To choose | Type
--- | ---
Odd parity, conversion, and no translation | C
Even parity, no conversion, and no translation | E
Odd parity, no conversion, and translation | T
Even parity, no conversion, and translation | ET
Data compression | COMP
No data compression | NOCOMP

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 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 Main Menu.
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 on page 107):

<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)  
  $SORTPARM data set name (below)  
```
2 Specify the options for SORTWORK data sets:
   a In the **Number of SORTWORK Data Sets** field, type the number of SORTWORK data sets.
   b In the **SORTWORK unit name** field, type the name of the unit for SORTWORK data sets.

   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.

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 100, specify the default primary and secondary sizing 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 `$ORTPARM 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 Main Menu.

---

**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>

**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 on page 111).

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>(CATALOG MANAGER only) 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 on page 112).

Figure 40: Using the &GDG symbolic variable

```
COMMAND ===>
```

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 . . . . . . . . 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)
```

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 on page 112).

Figure 41: Data set names resolved with the &GDG symbolic variable

```
/*  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,
```

112 CATALOG MANAGER for DB2 User Guide
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 “Setting the JCL options for static data sets” on page 100) 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.

- **(CATALOG MANAGER only)** If you want to execute the HDDDL line command, enter a valid Management Class. CATALOG MANAGER uses the Management Class instead of the SORTOUT Unit name. CATALOG MANAGER cannot sort HDDDL on tape.

4 If you typed **N** in the **Data set sizing option** field in Step 2 on page 100, 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.

This option is not applicable if you choose to dynamically allocate copy or unload data sets.

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.

   **Note**
   
   (CATALOG MANAGER only) CATALOG MANAGER cannot sort HDDL on tape. If you want to execute the HDDL line command and have specified TAPE for the SORTOUT Unit name (sort output), you must edit the Product Options File (POF) or create a new POF. The device where the data set is saved is defined according to the first POF variable, in the following order, that contains a valid value:

   1 ACTWRK_MGMTCLASS
   2 ACTWRK_STORCLASS
   3 ACTWRK_DATACLASS
   4 ACTWRK_UNIT

   For details, see “Creating a user POF” on page 127 or “Updating a user POF” on page 129.

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.
For more information about the TEMPLATE descriptors, see *ALTER and CHANGE MANAGER for DB2 User Guide, Volume 2* and the -COPY worklist command in the *ALTER and CHANGE MANAGER for DB2 Reference Manual*.

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 on page 114 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 104.

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 Main Menu.

---

**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 on page 116):

<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>
</tbody>
</table>
From this menu | Select this item and press Enter
---|---
JCL Generation Update - Main Menu | Generation Data Group Options (GDGs)

**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.

*Note*
If you type NONE, the DCB=**model dataSetName** is omitted from the JCL for the data set.

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

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

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

AJXODBG JCL GENERATION DEBUGGING,DISPLAY AND EXECUTION OPTIONS UPDATE ----------
COMMAND ===> Type data and press Enter.

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

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

**Note**

BMC recommends that you include the comments if you suspect that the JCL was generated incorrectly and you need to send documentation to Customer Support. If you want to reduce the number of lines of JCL, do not include the comments.

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 on page 118** 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>H</td>
<td></td>
<td>Uses the high relative-byte address (RBA)</td>
</tr>
<tr>
<td>M</td>
<td></td>
<td>Uses multiple objects to size one data set (for example, SYSUTs)</td>
</tr>
<tr>
<td>N</td>
<td></td>
<td>Indicates that the data set could not be sized because statistics could not be found</td>
</tr>
<tr>
<td>O</td>
<td></td>
<td>Uses VSAM object sampling</td>
</tr>
<tr>
<td>S</td>
<td></td>
<td>Uses the following formula to calculate the SORTWK size: <em>(work space * 2) / number of SORTWK data sets</em></td>
</tr>
<tr>
<td>W</td>
<td></td>
<td>Warns that the sizing might be inaccurate</td>
</tr>
</tbody>
</table>

**Figure 44 on page 118** shows example comments in the Execution JCL.

**Figure 44: Execution JCL with comments**

```c
//** ERDDN OUTPUT DD STATEMENTS
//**
//* N = CAN'T SIZE DATA SET FOR DD SYSER001, DEFAULTS USED BECAUSE
//* NO BMCSTATS WERE FOUND FOR TB ACMX01.T_X01PS
//SYSER001 DD DSN=RDAMCG3.MG1217D.STEP1.SYSER001,
// DISP=(NEW,CATLG,CATLG),
// SPACE=(CYL,(10,2),RLSE), ESTIMATE-B/N
// UNIT=SYSDA

//** SORT WORK DD STATEMENTS
//**
//* S = SORTWK SIZE IS (WORK SPACE * 2 / #SORTWKS)
//SORTWK01 DD UNIT=SYSDA,
// SPACE=(CYL,(1,1)), ESTIMATE-C/S
// DISP=(NEW,DELETE)
```

1. 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.

2. In the **Suppress comments in JCL** field, type **Y** or **N** to specify whether to suppress all comments in the generated JCL.
Note
If you chose to include either the statistics comments or the variable comments in Step 2 on page 117, 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 Main Menu.

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.

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 on page 120):
From this menu | Select this item and press Enter
--- | ---
CATALOG MANAGER Primary Menu panel | CATALOG MANAGER options processing
Options | Y at Edit JCL Generation options
JCL Generation Options | Y at Edit JCLgen options
JCL Generation Update - Main Menu | Utility Options
Utility Options | BMC Utility Option Module Names

---

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

```
---------- JCL GENERATION BMC UTILITY OPTION MODULE NAMES UPDATE ----------
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 Main Menu.

---

**Setting the online reorg options**

Use the Online Reorg Utility Options panel to specify the options for reorganizing table spaces.

**Note**

CATALOG MANAGER and DASD MANAGER PLUS currently do 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 on page 121):
From this menu | Select this item and press Enter
--- | ---
CATALOG MANAGER Primary Menu | CATALOG MANAGER options processing
Options | Y at Edit JCL Generation options
JCL Generation Options | Y at Edit JCLgen options
JCL Generation Update - Main Menu | Utility Options
Utility Options | Online Reorg Options

Figure 46: Online Reorg Utility Options panel

```
COMMAND ===> ONLINE REORG UTILITY OPTIONS
BMCREORG XBMID: XBM
REORG MAPPING TABLE: J234.JFLTBMAB
REORG MAPPING DATABASE: XXX
NOTE: The mapping table full length cannot exceed 72 characters
NOTE: The mapping database full length cannot exceed 8 characters
```

2 In the **BMCREORG XBMID** field, specify the BMC EXTENDED BUFFER MANAGER (XBM) subsystem (SSID) that the 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.

**Note**

(DB2 Version 11 and later) If you want to use the system default values of IBM REORG, (not define a mapping table or a mapping database) for online reorg (SHRLEVEL CHANGE), perform the following steps:

1 On the Online Reorg Utility Options panel, do not specify a mapping database name.

2 On the Online Reorg Utility Options panel, do not specify a mapping table name.

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

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

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.

If you are specifying a mapping table for DB2 Version 11 and later, ensure that the length of the LRSN column is appropriate for the tablespace that is being reorganized.

4 (DB2 Version 11 and later) In the REORG MAPPING DATABASE field, specify the name of the default database that the IBM REORG utility uses when it implicitly creates a mapping table.

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

---

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 on page 123):

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

**Figure 47: PROC and STEP Names panel**

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>PROCNAME</th>
<th>PROCSTEP</th>
<th>FUNCTION</th>
<th>PROCNAME</th>
<th>PROCSTEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC REORG</td>
<td>BMCREORG</td>
<td>RSTEP</td>
<td>DSNUTILB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC COPY</td>
<td></td>
<td></td>
<td>TSO BATCH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC LOAD</td>
<td></td>
<td></td>
<td>DSN1COPY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC UNLOAD</td>
<td></td>
<td></td>
<td>IDCAMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC RECOVER</td>
<td></td>
<td></td>
<td>IEFBR14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC CHECK</td>
<td></td>
<td></td>
<td>BMC STATS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC TRIG</td>
<td></td>
<td></td>
<td>BMC CPRS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMC UPRS</td>
<td></td>
<td></td>
<td>BMC STOP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER DEF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. In the *Use JCL Procedures (PROCS) for standard JCL?* field, type `Y` or `N` to specify whether to generate a PROC name instead of the `EXEC PGM=` statement in non-worklist JCL.

If you specify `Y`, you must provide a PROC name and a STEP name for the utilities for which you want to generate JCL. You are responsible for creating the PROC.

**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 96). 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.

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

The AJXPSETV SLIB contains SET statements that assign SLIB variables to parameters. If you specify `Y`, JCL Generation generates SET statements for the parameters in the JCL.
Tip
You can add parameters or change the variable names in the AJXPSETV SLIB. If you change any values in the AJXPSETV SLIB member, you must compile and link the member to your current load library. For sample compile JCL, refer to member AJXCOMPS in the HLQ.BMCCNTL data set.
You can also specify to generate SET variables in the JCL in the PROC_GEN_SET_VAR POF keyword in your POF.

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

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_BMCREORG_NAME =
- PROC_BMCSTATS_NAME =
- PROC_BMCSTOP_NAME =
- PROC_BMCTRIIG_NAME =
- PROC_BMCUNLOAD_NAME =
- PROC_BMCUPRS_NAME =
- PROC_DSNUTILB_NAME =
- PROC_DSN1COPY_NAME =
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_IDCAMS_STEP** =
- **PROC_IEFBR14_STEP** =
- **PROC_TSO_STEP** =
- **PROC_USER_DEF_STEP** =

5 Press END to save your changes, and to return to the JCL Generation Update - Main Menu panel Main Menu.
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 on page 126):

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

Figure 48: JCL Generation IBM Utility Dynamic Data Set Options Update panel

```
COMMAND ===>
IBM Utility Dynamic Data Set Options
Enter data and press Enter.
  Enter External Template Data Set Name below: ...
  Enter External Listdef Data Set Name below: ...
```

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

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

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

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

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

To define or modify the values in your ISPF profile and a user POF, you can use the Options panels of the JCL Generation component.

Use the User Defined Variables panel to specify character variables. Each variable has a corresponding symbolic variable that you can use in job cards or data set prefixes.

To set the JCL options for user-defined variables

1. Use the following menu selections to display the User Defined Variables panel (Figure 49 on page 127):

<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

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 Main Menu.

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.

**To create a user POF**

1. Use the following menu selections to display the JCL Generation Product Options File (POF) Functions panel (Figure 50 on page 128):

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

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

```
COMMAND ===> TYPE DATA AND PRESS ENTER. PRESS PF3 OR END TO RETURN TO THE MAIN PANEL.

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

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

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

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

You can update a user POF by using the options panels or by directly editing the file.

**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.BMCCLIB 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 *HLQBMCCLIB* 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.

1. Follow the steps in “Creating a user POF” on page 127 to create an additional POF.

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

   **Note**
   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**

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**

1. 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

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: Reusing an existing POF

3. Enter the names of the data set and member for the existing POF (Figure 51 on page 133).

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

   Figure 52: Specifying the name of the existing POF
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 on page 134), 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.

Figure 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 on page 134), the DBA can uphold your site’s naming standards.

Figure 54: Changing the SLIB variable for the copy data set in AJX#DSNS

```
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 *ALTER and CHANGE MANAGER for DB2 User Guide*, Volume 2.

### Adding steps to the JCL

In CATALOG MANAGER or DASD MANAGER PLUS, 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

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>
</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: JCL Generation Jobcard Options Update panel**

```
------------------- 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:
//JOBCDBDC JOB (&ZACCTNUM), 'PGMR',
// CLASS=A, MSGCLASS=X, MSGLEVEL=(1,1),
// NOTIFY=&USERID
/**
 ***
Jcllib . . . . . . SKH.INCLUDE.JCLLIB
Sysexec. . . . . . (See JCL Reference for valid options)
Region size . . . 0M
Time parameter . . . (See JCL Reference for valid options)
System MLIB. . . . ISP.SISPMENU
Runtime HLQ. . . . AEX.QA0101
LLQ . . . . . . . . BMC
ULLQ . . . . . . . (Leave blank if using runtime enablement)
```

When you use the JCLLIB option, the JCL Generation component generates the following statement in the JCL:

**Figure 56: 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: 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:
SYNDELETE . . . N (Y/N) BINDFAIL . . . N (Y/N)
HASHFAIL . . . N (Y/N) HASHWARNRC . . . (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: Including JCL between steps**

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

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

  The JCL Generation component generates the following statement in the JCL at the end of the job.

  **Figure 59: Including JCL at the end of the jobs**

  ```
  //*--------------------------------
  //* END OF JOB INCLUDE MEMBER
  //*--------------------------------
  // INCLUDE MEMBER=JOBEND
  ```

When you run the JCL, the members that you specified are expanded, as shown in the following segment of a JES log.

**Figure 60: Expanded INCLUDE members**

```plaintext
3 // INCLUDE MEMBER=STEPEND
XX** CUSTOMIZED PROGRAM TO DO SOMETHING BETWEEN STEPS
4 XXXSTEPEND EXEC PGM=STPCOND,PARM=&SYSUID,COND=EVEN
```
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.

Related Information

- “TEMPLATE and LISTDEF control statements” on page 291

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.BMCCNTL library.
This member contains all of the correct values for your installation. The contents vary among versions of CATALOG MANAGER. Figure 61 on page 139 shows a portion of an ACTCOMND file.

**Figure 61: Sample ACTCOMND file**

```plaintext
EJECT
  COPY $ACTCEQU
  $ACTCMD TYPE=DSECT
ACTCOMND CSECT
ACTCOMND AMODE 31
ACTCOMND RMODE ANY
  *ACTIVATE
    $ACTCMD ACTIVATE,@ACT,LOAD=ACTNCFMM,HELP=ACTHACTI,LOG=YES, CAT=YES,DB2CMD=YES,
    D2VMIN=DB2V9ENFM,
    OBJECTS=(NP+PR)
  *LIST AL (ALIASES)
    $ACTCMD AL,#AL,LOAD=ACTXXXXX,HELP=ACTHLSAL,LIST=YES,CAT=YES, CAT=YES,DB2CMD=YES,
    D2VMAX=DB2V8,
    OBJECTS=(CX+DB+FN+IM+IX+MQT+PR+SE+SG+TB+TS+UN)
  *ALTER (ALTER)
    $ACTCMD ALTER,#ALT,LOAD=ACTNCFMM,HELP=ACTHALTR,LOG=YES, CAT=YES,DB2CMD=YES,
    D2VMAX=DB2V9,
    OBJECTS=(CX+DB+FN+IM+IX+MQT+PR+SE+SG+TB+TS+UN)
  *ALTER (ALTER)
    $ACTCMD ALTER,#ALT,LOAD=ACTNCFMM,HELP=ACTHALTR,LOG=YES, CAT=YES,DB2CMD=YES,
    D2VMIN=DB2V9ENFM
    OBJECTS=(CX+DB+FN+IM+IX+MQT+NP+PR+SE+SG+TB+TS+UN)
  *LIST AP (AUDITPOLICIES)
    $ACTCMD APO,#APO,LOAD=ACTXXXXX,HELP=ACTHAPOL,LIST=YES,CAT=YES,
    OBJECTS=(AP)
  *APPLY
    $ACTCMD APPLY,#APP,LOAD=ACTNCFMM,HELP=ACTHAPLY,CAT=YES, WFEK=YES
```

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.
To set up the table

1. Copy HLQ.CNTL(ACTCOMNU) to HLQ.UBMCCNTL.

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 ACTCOMND 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.UBMCCNTL 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 143.

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 on page 140) to work like the DISPLAY UTILITY command (Figure 63 on page 140). You can replace the format of the STATUS command with that of the DISPLAY UTILITY command.

   **Figure 62: STATUS command**

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

   **Figure 63: DISPLAY UTILITY command**

   ```
   *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 on page 141.

**Figure 64: New STATUS command**

```
*DISPLAY STATUS
$ACTCMD_STATUS,#DSU,LOAD=ACTXXXXX,HELP=ACTHKDUT,CAT=YES, X 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.UBMCCNTL 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.

4. Run the USRCOMND job in the HLQ.JCL or HLQ.UBMCCNTL 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 143.

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 143.

3. Write the CLIST.

   For more information, see “Commands table syntax and parameters” on page 143.

4. For more information, see “Commands table syntax and parameters” on page 143. Run the USRCOMND job in the HLQ.JCL or HLQ.UBMCCNTL data set to compile and link the 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.

To disable a command

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.UBMCCNTL data set to compile and link your user commands table.
Commands table syntax and parameters

This topic explains the syntax for entries in the commands table.

The syntax is as follows:

```
*commandName objectType (expandedName)
$ACTCMD commandName, #commandEquivalent, parameterName=value.
```

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commandName</td>
<td>Name of the command that appears in the Commands List panel</td>
</tr>
<tr>
<td>objectType</td>
<td>Object’s two-, three-, or four-character abbreviation</td>
</tr>
<tr>
<td>expandedName</td>
<td>Alternate or full name of the command (for example, ALIASES)</td>
</tr>
<tr>
<td>commandEquivalent</td>
<td>Characters or numbers that represent a command</td>
</tr>
<tr>
<td></td>
<td>For example, #COL or 081 can represent the COLSTATS command. For a list</td>
</tr>
<tr>
<td></td>
<td>of valid characters and numbers, see the $ACTCEQU member in the</td>
</tr>
<tr>
<td></td>
<td>HLQ.BMCMAC library.</td>
</tr>
<tr>
<td>parameterName</td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>HLQ.BMCMAC library. Also, some of the parameters in the member are</td>
</tr>
<tr>
<td></td>
<td>described in this section.</td>
</tr>
<tr>
<td>value</td>
<td>Valid value for the parameter</td>
</tr>
</tbody>
</table>

You can use the following parameters with $ACTCMD:

---

**Note**

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

---

&CLIST

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

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

Specifies the maximum (latest) version of DB2 for which the command is valid.

&DB2MIN

Specifies the minimum (earliest) version of DB2 for which the command is valid.

&HELP

Specifies which Help panel to display when the command is selected. The default is ACTHCMDS.

&LOAD

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

&LOG

Indicates whether any CATALOG MANAGER commands, DML commands, and DB2 TERM UTILITY commands are recorded 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

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

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

&NOSERVER

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

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

Lists the two-character codes for the object types to which the command applies. See the tables in “Selecting an Action” on page 34 to 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

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

&PLAN

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.BMCCNTL(ACTCOMNU) for the previous release to HLQ.BMCCNTL(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.UBMCCNTL library. Copy the modifications that you made to HLQ.UBMCCNTL(ACTCOMNU) to member ACTCOMNU in the latest release of the HLQ.UBMCCNTL library.
     
     **Note**
     This step assumes that you have made your changes in the HLQ.UBMCCNTL library.
   - If no differences exist between the two members, copy HLQ.UBMCCNTL(ACTCOMNU) for the previous release to the latest release of the HLQ.UBMCCNTL library.

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

4. Run the USRCOMND job in the HLQ.JCL or HLQ.UBMCCNTL 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.
Development aids for user commands

The following table lists the members that are shipped with the product to aid in the development of your own CATALOG MANAGER commands.

Table 23: Customizable code for building user 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>$ACTEXC</td>
<td>Macro DSECT for user program variables</td>
</tr>
<tr>
<td>$ACTULOG</td>
<td>Macro for user logging</td>
</tr>
<tr>
<td>$ACTLISTC</td>
<td>Sample CLIST that displays IDCAMS LISTCAT output for a given data set</td>
</tr>
<tr>
<td>$ACTVARS</td>
<td>Sample CLIST that shows how the variables are set when you execute a CLIST from a list. 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.</td>
</tr>
<tr>
<td>USRCOMND</td>
<td>JCL to assemble and link the ACTCOMND source code for the commands table</td>
</tr>
</tbody>
</table>

Command program parameters

A user command program is invoked using standard operating system calling conventions:

- Save area in register 13
- Address in register 15
- Return in register 14

Register 1 points to the user program parameter list. The macro $ACTEXC provides the DSECT of the parameter list. This macro is in the HLQ.BMCMAC library.

Table 24 on page 148 lists the key parameters.
Table 24: Key parameters for a user command program

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXCRC</td>
<td>ISPF END function to execute on return</td>
</tr>
<tr>
<td></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 line</td>
</tr>
<tr>
<td>EXCCOBJL</td>
<td>Object list type where the command is valid</td>
</tr>
<tr>
<td>EXCCOBJC</td>
<td>Object type found in the command text if PARSE=YES</td>
</tr>
<tr>
<td></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 25 on page 148.

Table 25: Parameters for use with the $ACTULOG macro

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pointer</td>
<td>Pointer to a log text field</td>
</tr>
<tr>
<td></td>
<td>The field should be in the format $LLtext where $LL$ is the length of the</td>
</tr>
<tr>
<td></td>
<td>field including itself. The $text$ can be whatever you want put in the log.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| OPT | The logs to which you want to write  
The options are as follows:  
- A - Audit Log  
- S - Session Log (default)  
- B - both the Audit Log and the Session Log |
| FUNC | Address of a field containing the 16-byte function to log  
It does not need to match the command name. If this parameter is not used, blanks are logged. |
| RC | Register containing the return code to be logged  
If this parameter is not used, X'00' is logged as the return code. |
| TYPE | Address of a 10-byte value to be logged as the object type  
If this parameter is not used, blanks are logged. |
| QUAL | Address of an 8-byte value to be logged as the object name qualifier  
If this parameter is not used, blanks are logged. |
| NAME | Address of an 18-byte value to be logged as the object name  
If this parameter is not used, blanks are logged. |

### 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.
If you invoke the CLIST command on an object type that is not supported, CATALOG MANAGER places an appropriate message in the ISPF EXCOFNM0 variable. If you invoke the CLIST command on an invalid CLIST name, CATALOG MANAGER issues a message and continues to execute.

Table 26 on page 150 lists the object types and their corresponding name fields.

Table 26: 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) 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 number</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 name</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Image copy</td>
<td>IC</td>
<td>NA</td>
<td>DB name</td>
<td>TS name</td>
<td>Date</td>
<td>EXCOFNDB (DB name) EXCOFNTS (TS name) EXCOFNM4 (time) EXCOFNMID (date)</td>
</tr>
<tr>
<td>Index</td>
<td>IX</td>
<td>NA</td>
<td>Creator</td>
<td>Name</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>IX partition</td>
<td>IP</td>
<td>NA</td>
<td>IX creator</td>
<td>IX name</td>
<td>Partition</td>
<td>NA</td>
</tr>
<tr>
<td>Key column</td>
<td>KC</td>
<td>Column 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) 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) EXCOFNM5 (contoken) EXCOFNM6 (location)</td>
</tr>
</tbody>
</table>
Table 27 on page 151 describes the ISPF variables that you can access from user-written commands or CLISTs.

Table 27: ISPF variables for user commands or CLISTs

<table>
<thead>
<tr>
<th>Variable</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;VCAT</td>
<td>8</td>
<td>High-level qualifier of the VSAM catalog (VCAT)</td>
</tr>
<tr>
<td>&amp;SSID</td>
<td>4</td>
<td>DB2 subsystem Identifier</td>
</tr>
<tr>
<td>&amp;ACTSRVR</td>
<td>16</td>
<td>Active server</td>
</tr>
<tr>
<td>Variable</td>
<td>Length</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>&amp;ACTCOLID</td>
<td>18</td>
<td>Active collection identifier</td>
</tr>
</tbody>
</table>

## CATALOG MANAGER tables

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

**Note**

To prevent accidental updates to the data in Table 28 on page 152, set the status of the table spaces to **Read Only**.

### Table 28: CATALOG MANAGER tables

<table>
<thead>
<tr>
<th>Table name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMCACT vr.ATTR</td>
<td>Contains information about attributes</td>
</tr>
<tr>
<td>BMCACT vr.ATTR_VAL</td>
<td>Contains the valid values for each attribute</td>
</tr>
</tbody>
</table>
| BMCACT vr.AUDIT_LOG   | Contains an entry for each DDL statement that is executed by CATALOG MANAGER to modify the DB2 catalog
For more information, see “The DDL Audit Log” on page 350. |
| BMCACT vr.CRS_VAL     | Contains cross-validation between attributes |
| BMCACT vr.DLG         | Contains dialog-specific information |
| BMCACT vr.DLG_ATTR    | Contains the attributes to be displayed for each dialog |
| BMCACT vr.EDITOR_USERS| Contains one row for each data editing session in progress
A row is inserted when data is fetched and deleted when the editing session ends. All rows that reference a particular user ID are deleted when that user ID begins another editing session.
If an editing session ends abnormally, the applicable row might not be deleted from the table. Users with SYSADM authorization can delete all rows from the table by using an option on the Edit and Browse Options panel.
For more information about the Clear Editor Users Table option, see “Setting options for browsing data” on page 216 |
| BMCACT vr.FCRS        | Contains the Filter Combo Result Table |
| BMCACT vr.GMAP        | Contains the grid mapping table |
| BMCACT vr.MSG         | Contains the informational messages |
### Table name | Description
--- | ---
BMCACT `vr.RECOVERY_LOG` | Contains an entry for each object that is dropped and each privilege that is revoked when the DROP statement is executed within CATALOG MANAGER and the Drop Recovery option is set to Y. For more information, see “The Drop Recovery Log” on page 351.

BMCACT `vr.SEARCH_VARS2` | Contains the values for variables that were entered in SEARCH statements and saved. For more information, see “Using host variables in a search” on page 185.

BMCACT `vr.SESSION_LOG` | Contains an entry for each CATALOG MANAGER action for which logging was requested. For more information, see “The Session Log” on page 346.

BMCACT `vr.SQL_TABLE` | Contains SQL statements that have been saved. For more information, see “Using the SQL_Table” on page 207.

BMCACT `vr.VIEW` | Contains the initial view for each utility that is supported.

---

**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.

“Accessing other DB2 subsystems” on page 155 explains how to change the default SSID attachment in order to work with the catalog on another SSID.
Where to go from here
Accessing other DB2 subsystems

A CATALOG MANAGER session is always attached to a single DB2 subsystem (SSID) on the same z/OS system.

You might connect the SSID to another single SSID in order to query its catalog. CATALOG MANAGER provides the following features for you to substitute either or both of these SSIDs without closing your CATALOG MANAGER session:

- Use the DB2 Attach feature to modify the attachment between CATALOG MANAGER and an SSID.
- Use the DB2 Connect feature to establish or modify a connection between the attached SSID and another SSID.

Using the DB2 Attach feature

When you start CATALOG MANAGER, the session is attached to a single DB2 subsystem as defined in the invoking BMCDB2 CLIST.

The DB2 Attach feature enables you to end the attachment between CATALOG MANAGER and the current SSID, and then establish an attachment to a different SSID on the same z/OS system. You are still able to use all CATALOG MANAGER features.

DB2 Attach has the following requirements:

- CATALOG MANAGER must be installed on the target z/OS system at the same maintenance level as the product instance from which you are connecting.
- The target SSID must be on the same z/OS system as CATALOG MANAGER.
- The DB2 Call Attach Facility (CAF) must be installed.

When you attach to a different subsystem, the installation options and profile that were invoked on your original CATALOG MANAGER session are still in effect. If you make changes to your options while attached to another DB2 subsystem, the
changes apply to your original session as well. Figure 65 on page 156 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

Use the following procedure 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: Change Access panel showing DB2 attach specifications**

<table>
<thead>
<tr>
<th>Command</th>
<th>Change Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display quick connections list</td>
<td>N (Y/N)</td>
</tr>
<tr>
<td>Display product plans and connections table</td>
<td>N (Y/N)</td>
</tr>
</tbody>
</table>

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>10</td>
<td>DEFF</td>
<td>Call Attach</td>
</tr>
<tr>
<td>Plan</td>
<td>ACT101DM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server</td>
<td></td>
<td></td>
<td>Current Server</td>
</tr>
<tr>
<td>SQLid</td>
<td>RDACRJ2</td>
<td></td>
<td>SOLid on Server</td>
</tr>
<tr>
<td>Server SSID</td>
<td></td>
<td></td>
<td>SSID of Server</td>
</tr>
<tr>
<td>Collection</td>
<td></td>
<td></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.

**Note**

If the SSID that you select is incompatible or otherwise unavailable, a warning message is displayed in the short message area of the Change Access panel.

4 Press **END** to display the Primary Menu panel.

The newly attached SSID is displayed at the bottom of the panel.

### Attaching to an SSID or server by using the connection selection list

You can also select values for the **SSID**, **Plan**, **Server**, and **Collection** fields from the connection selection list that is created during the installation of CATALOG MANAGER.
1 From the Primary Menu panel, enter CONNECT (CON) on the Command line.

   The Change Access panel is displayed (see “Attaching CATALOG MANAGER to a specified SSID” on page 156).

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, which is 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.

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.
**Tip**
As an alternative to this procedure, you can type `SET QUALIFIER AliasQualifier` on the Command line to switch catalog access.

---

**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.

1. Run one 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.

2. After you restore the attachment, refresh the product options file (POF).

---

**Related Information**

- "Refreshing the initial POF" on page 131
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.
- 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 164.
- 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 manipulate objects” on page 165.

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 support the following commands:

- SPACE and STATS commands that invoke DASD MANAGER PLUS
- The DSN command DCLGEN
- The DSN commands BIND, REBIND, or FREE for PLAN, however these commands are supported for PACKAGES
Figure 67 on page 161 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.

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.

   **Tip**

   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.

3. In the Collection field, type the collection name for the catalog, catalog view, or catalog copy that you want to use.

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 on page 163). The collection ID that CATALOG MANAGER is using on the connected SSID is displayed in the Current Value column and in the short message area of the panel.

**Figure 68: Change Access panel after new connection**

<table>
<thead>
<tr>
<th>DB2V</th>
<th>Current Value</th>
<th>Change to</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSID</td>
<td>10 DEFF</td>
<td>ACT101DM</td>
<td>Call Attach</td>
</tr>
<tr>
<td>Server</td>
<td>91 DBDC</td>
<td>Current Server</td>
<td></td>
</tr>
<tr>
<td>SQLid</td>
<td>RDACRJ</td>
<td>SQLid on Server</td>
<td></td>
</tr>
<tr>
<td>Server SSID</td>
<td>DBDC</td>
<td>SSID of Server</td>
<td></td>
</tr>
<tr>
<td>Collection</td>
<td>ACT_QA1010</td>
<td>Direct/Indirect</td>
<td></td>
</tr>
</tbody>
</table>

6 Press END to display the Primary Menu panel.

The current connection is noted at the bottom of the panel.

**To specify the CONNECT command and parameters**

1 From the Primary Menu panel, enter CONNECT (CON) parameters on the Command line. Table 29 on page 163 defines the parameters.

**Table 29: CONNECT command parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
</table>
| **Location** | Identifies the remote DB2 SSID  
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). |
| **Collection** | *(optional)* Names the collection and package set for the remote DB2 SSID  
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 =). |
| **SQLID** | *(optional)* Specifies the current SQLID for the remote DB2 SSID  
The default value is the SQLID that the local DB2 SSID uses |
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 “Getting started with CATALOG MANAGER” on page 29.

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 131.

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

You can connect to a remote DB2 SSID from a location (LO) list.

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 201.

---

### Connecting to a remote SSID and issuing DB2 commands to manipulate 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.
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 “Getting started with CATALOG MANAGER” on page 29.

2. Connect to a remote SSID.

   For more information, see “Connecting to a specified SSID” on page 161.

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 201.

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.
To re-establish a saved connection

1. From the Connections list panel, select the connection from the displayed connections list. See Figure 69 on page 167.

   **Note**
   The connections list does not duplicate saved connections. Only the first of identical saved connections is displayed on the connections list.

The data on the Connections List panel (Figure 69 on page 167) includes the following information:

- Connections that were saved by all users before you started the current CATALOG MANAGER session
- New, unique connections that you have established during the current CATALOG MANAGER session

   **Note**
   You cannot see new connections made by other users during their current sessions.

### Figure 69: Connections List panel

```
DEFF-S ----------------------  Connections List  ------------- Row 1 to 5 of 5
Command ===>                                                  Scroll ===> CSR  01

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Quickname</th>
<th>Location</th>
<th>Collection</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DEFF-DEC</td>
<td>ACT_QA1010</td>
<td>RDAPXB</td>
<td></td>
</tr>
<tr>
<td>DBDC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFF-D</td>
<td>DECA</td>
<td>ACT_DA1010</td>
<td>MVSJXL2</td>
<td></td>
</tr>
<tr>
<td>DEFF-D</td>
<td>DECK</td>
<td>ACT_DA1010</td>
<td>MVSJXL1</td>
<td></td>
</tr>
<tr>
<td>DBDC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEFF-DEC</td>
<td>DECA</td>
<td>ACT_DA1010</td>
<td>MVSJXL2</td>
<td></td>
</tr>
<tr>
<td>DEFF-D</td>
<td>DECK</td>
<td>ACT_DA1010</td>
<td>MVSJXL1</td>
<td></td>
</tr>
<tr>
<td>DEFF-DEC</td>
<td>DECA</td>
<td>ACT_DA1010</td>
<td>MVSJXL2</td>
<td></td>
</tr>
<tr>
<td>DEFF-D</td>
<td>DECK</td>
<td>ACT_DA1010</td>
<td>MVSJXL1</td>
<td></td>
</tr>
<tr>
<td>DBDC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

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.
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.

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.

To re-establish the connection while bypassing the Connections List panel

If you know the quickname of a saved connection, you can re-establish the connection while bypassing the Connections List panel.

1. 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 on page 167.

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 Customization Guide.

To view the connections table

1. From the Primary Menu panel, enter CONNECT (CON) on the Command line.

   The Change Access panel is displayed.
On the Command line, enter CONTAB.

The Connections Table panel is displayed.

**Figure 70: Section of Connections Table panel**

<table>
<thead>
<tr>
<th>SSID</th>
<th>Plan</th>
<th>Collection</th>
<th>Nickname</th>
<th>Location</th>
<th>LocID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CK</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>CK</td>
<td></td>
</tr>
<tr>
<td>DEDQ</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDQ</td>
<td></td>
</tr>
<tr>
<td>DEDZ</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDZ</td>
<td></td>
</tr>
<tr>
<td>DEDX</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDX</td>
<td></td>
</tr>
<tr>
<td>DEDV</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDV</td>
<td></td>
</tr>
<tr>
<td>DEDW</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDW</td>
<td></td>
</tr>
<tr>
<td>DEDA</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEDA</td>
<td></td>
</tr>
<tr>
<td>DECA</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DECA</td>
<td></td>
</tr>
<tr>
<td>DECI</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DECI</td>
<td></td>
</tr>
<tr>
<td>DECS</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DECS</td>
<td></td>
</tr>
<tr>
<td>DEEG</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEEG</td>
<td></td>
</tr>
<tr>
<td>DECC</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DECC</td>
<td></td>
</tr>
<tr>
<td>DEES</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEES</td>
<td></td>
</tr>
<tr>
<td>DEEB</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEEB</td>
<td></td>
</tr>
<tr>
<td>DEFU</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEFU</td>
<td></td>
</tr>
<tr>
<td>DEFV</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEFV</td>
<td></td>
</tr>
<tr>
<td>DEEU</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEEU</td>
<td></td>
</tr>
<tr>
<td>DEGA</td>
<td>D</td>
<td>ACT101DM</td>
<td>ACT_QA1010</td>
<td>DEGA</td>
<td></td>
</tr>
</tbody>
</table>

Table 30 on page 169 describes the columns on the Connections Table panel.

**Table 30: 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 0 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.

The following types of DB2-identifiers are available:

- 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.

To use the CONNECT command

1 Enter the CONNECT command followed by the DB2-identifier on the Command line of the Primary Menu panel.

For example, you can enter CONNECT DBDB-DBBF on the Command line of the Primary Menu panel to establish the designated connection. In this example, DBDB-DBBF would be the nickname assigned to the connection.

Identification of 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 “Viewing the connections table” on page 168 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

An attempt to attach or connect can fail or produce an unexpected result.

To trouble shoot DB2 Attach or DB2 Connect failure

1 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. “Working with lists and searches” on page 173 describes how to generate object lists and use them to manage your DB2 catalog information.
Working with lists and searches

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.

This section explains how to customize list displays and specify qualifiers other than the object name. For details of how to generate object lists from the Primary Menu and from other object lists, see “Getting started with CATALOG MANAGER” on page 29.

For more information, view the Quick Course "Understanding List Processing."

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 201.

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

The following table lists the source objects from which you can generate mixed lists.

The instruction area of most list panels specifies whether the MX command is valid for that panel.

Table 31: Valid source objects for mixed lists

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>DB</td>
</tr>
<tr>
<td>Plan</td>
<td>PL</td>
</tr>
<tr>
<td>Storage group</td>
<td>SG</td>
</tr>
<tr>
<td>Synonym</td>
<td>SY</td>
</tr>
<tr>
<td>Table</td>
<td>TB</td>
</tr>
<tr>
<td>Table name</td>
<td>TN</td>
</tr>
<tr>
<td>Table space</td>
<td>TS</td>
</tr>
<tr>
<td>Table space set</td>
<td>TT</td>
</tr>
<tr>
<td>View</td>
<td>VW</td>
</tr>
</tbody>
</table>

Generating a mixed list

This procedure describes how to generate a mixed list of objects that are associated with a table space.

CATALOG MANAGER displays a mixed list of objects associated with the source table space. Where possible, objects are indented to show dependency, see Figure 72 on page 175.

To generate a mixed list

1. Generate a list of table spaces.

   For more details, see “Generating lists in CATALOG MANAGER” on page 45.
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: Table Space List panel to generate a mixed list

```
DEFF-R ------------------------ TABLESPACE LIST --------------- ROW 1 OF 841
Command ===>                                                  Scroll ===> CSR
01
CMD will show commands for this list. Type command and press ENTER
Lists: ACCTB AL BMCUHIST CA CL CO DB DS FK IC IM IS IX LK M0T MX NP OS PA PDD
LIKE %.QZU%
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----
QZRDTIM.QZUS0101  ASUQA     16 BP0       0    1        -1        A    E
MXQZUDAC.QZUS01AC ASUQA     0 BP0       4     1 233        A    E
QZUDAC.QZUS02AC  ASUQA     0 BP0       4     1 233        A    E
QZUDAC.QZUS03AC  ASUQA     4 BP0       1     1 5040        A    E
QZUDAC.QZUS04AC  ASUQA     4 BP0       0     1 244        A    E
QZUDAC.QZUS05AC  ASUQA     8 BP0       0     1 244        A    E
QZUDAC.QZUS06AC  ASUQA     16 BP0       0    1 244        A    E
QZUDAC.QZUS07AC  ASUQA     32 BP0       0    1 244        A    E
QZUDAC.QZUS08AC  ASUQA     64 BP0       0    1 244        A    E
QZUDAC.QZUS09AC  ASUQA     128 BP0       0   1 244        A    E
QZUDAC.QZUS10AC ASUQA     16 BP0       2    1 307K        A    E
QZUDAC.QZUS11AC ASUQA     4 BP0       0     1 130        A    A
QZUDAC.QZUS12AC ASUQA     16 BP0       0     2 146        A    A
QZUDAC.QZUS13AC ASUQA     0 BP0       4     1 1440       A    A
QZUDAC.QZUS14AC ASUQA     64 BP0       0     2 540        A    A
QZUDAC.QZUS15AC ASUQA     0 BP0       4     1 720        A    A
QZUDAC.QZUS16AC ASUQA     4 BP0       0     1 8221       A    A
```

3. Press Enter.

Figure 72: Mixed Object List panel generated from table space list

```
DEFF-R ------------------------ MIXED LIST --------------- ROW 1 OF 9
Command ===>                                                  Scroll ===> CSR
02
CMD will show commands for this list. Type command and press ENTER
Lists: ANY VALID LIST FROM THE SELECTED ROW
QUALIFIER: TABLESPACE=QZUDAC.QZUS01AC
Cmd  Obj Type   ObjInf   Tblspace Name of Object   Plan or Package
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
TS         0004     QZUS01AC
.TB        TCPT QZU.QZUT01_DACS01
..IX       PART QZU.QZUX01_DACS01T01
..IX       DPSI QZU.QZUX02_DACS01T01
..IX       PART QZU.QZUX03_DACS01T01
..IX       DUPS QZU.QZUX04_DACS01T01
..IX       PART QZU.QZUX05_DACS01T01
..IX       DPSI QZU.QZUX06_DACS01T01
..IX       UNIQ QZU.QZUX07_DACS01T01
```

Note

On the Object Use Options panel (see “Setting object use options” on page 81), you can specify whether to exclude synonyms, plans, and packages from mixed object lists.

You can issue utility and DDL commands against objects on a mixed list as you can on any other list. You can also generate additional secondary lists for objects on a mixed list. For a list of these object types and the lists that you can generate for them, type CMD after generating a mixed object list.
Combining lists

A combined list displays a single object type that is associated with multiple source objects in a list.

You can generate a combined list from either a level-one list or a secondary list.

Table 32 on page 176 shows the source object types and codes for which you can generate a combined list.

Table 32: Valid source objects for generating combined lists

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>AL</td>
</tr>
<tr>
<td>Column</td>
<td>CO</td>
</tr>
<tr>
<td>Database</td>
<td>DB</td>
</tr>
<tr>
<td>Image copy</td>
<td>IC</td>
</tr>
<tr>
<td>Index mixed</td>
<td>IM</td>
</tr>
<tr>
<td>Index</td>
<td>IX</td>
</tr>
<tr>
<td>Package</td>
<td>PG</td>
</tr>
<tr>
<td>Plan</td>
<td>PL</td>
</tr>
<tr>
<td>Partition</td>
<td>PT</td>
</tr>
<tr>
<td>Relation</td>
<td>RE</td>
</tr>
<tr>
<td>Storage group</td>
<td>SG</td>
</tr>
<tr>
<td>Table</td>
<td>TB</td>
</tr>
<tr>
<td>View</td>
<td>VW</td>
</tr>
</tbody>
</table>

Generating a combined list

Use this procedure to generate a combined list of packages that a plan can use.

1. Generate a list of plans.

   For more details, see “Generating lists in CATALOG MANAGER” on page 45.

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 on page 177). An asterisk in the Package column indicates that the plan can use all packages in the associated collection.

**Figure 73: Combined package list**

<table>
<thead>
<tr>
<th>DEFF-R</th>
<th>PACKLIST LIST</th>
<th>ROW 1 OF 2191</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMD will show commands for this list. Type command and press ENTER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lists: PG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUALIFIER: PLAN=ALL FROM LAST LIST</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>Seq</th>
<th>Location</th>
<th>Collection Id</th>
<th>Package</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>BMCACT</td>
<td>*</td>
<td>2010-06-30-14.11.59.323515</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>BMSCC</td>
<td>*</td>
<td>2010-06-30-14.11.59.323515</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>ACKDEV</td>
<td>*</td>
<td>2010-05-27-11.25.47.920980</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>BMSCC</td>
<td>*</td>
<td>2010-05-27-11.25.47.920980</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>ACKESHQ</td>
<td>*</td>
<td>2010-07-23-16.11.33.150955</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>ACKAO</td>
<td>*</td>
<td>2010-09-03-14.20.23.271151</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>BMSCC</td>
<td>*</td>
<td>2010-09-03-14.20.23.271151</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>ACKWKCT</td>
<td>*</td>
<td>2010-09-28-14.25.48.972588</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>BMSCC</td>
<td>*</td>
<td>2010-09-28-14.25.48.972588</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>*</td>
<td>ACMA1PDM</td>
<td>*</td>
<td>2010-08-24-18.13.08.091557</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>*</td>
<td>ALUA1PDM</td>
<td>*</td>
<td>2010-08-24-18.13.08.091557</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>*</td>
<td>AUCA1PDM</td>
<td>*</td>
<td>2010-08-24-18.13.08.091557</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>AJX_QA1010</td>
<td></td>
<td>2010-08-24-18.13.08.091557</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>AEX_QA1010</td>
<td></td>
<td>2010-08-24-18.13.08.091557</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>SKH_DEV1010</td>
<td></td>
<td>2010-08-24-18.13.08.091557</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>ACS_QA1010</td>
<td>*</td>
<td>2010-08-24-18.13.08.091557</td>
<td></td>
</tr>
</tbody>
</table>

### 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 45.

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.

**Note**

To review procedures for issuing commands against listed objects, see “Issuing Wait-for-Enter commands against multiple objects” on page 55.
Using SEARCH to generate lists based on object attributes

In addition to generating lists based on object names (LIST command), you can use SEARCH to generate lists that match more specific variables.

**To use SEARCH to generate lists**

1. Enter the SEARCH command in one of the following ways:
   - On the Primary Menu panel or a list panel, on the **Command** line type `SEARCH(SEA) objectCode Qualifier`.
   - 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).
     - In the **Qualifier** field, type the qualifier (wild cards are accepted)

**Note**
The SEARCH command uses dynamic SQL. Individual users are required to have SELECT authority on any tables that are referenced when the SEARCH command is issued.

For more information, view the Quick Course "Searching the DB2 Catalog."

**Valid objects for searches**

The following table lists the source objects for which you can perform searches.

<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</td>
<td>CK</td>
</tr>
<tr>
<td>Check dependent</td>
<td>CD</td>
</tr>
<tr>
<td>Object type</td>
<td>Code</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Column</td>
<td>CO</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>Dependencies</td>
<td>DP</td>
</tr>
<tr>
<td>Environments</td>
<td>EN</td>
</tr>
<tr>
<td>IP addresses for a given location (DDF)</td>
<td>IL</td>
</tr>
<tr>
<td>Index</td>
<td>IX</td>
</tr>
<tr>
<td>IP name</td>
<td>IN</td>
</tr>
<tr>
<td>Java paths</td>
<td>JT</td>
</tr>
<tr>
<td>Key -targets of extended indexes</td>
<td>KT</td>
</tr>
<tr>
<td>Location a</td>
<td>LO</td>
</tr>
<tr>
<td>LU mode a</td>
<td>LM</td>
</tr>
<tr>
<td>LU mode select a</td>
<td>LS</td>
</tr>
<tr>
<td>LU name a</td>
<td>LU</td>
</tr>
<tr>
<td>Materialized query table</td>
<td>MQT</td>
</tr>
<tr>
<td>Native SQL procedure</td>
<td>NP</td>
</tr>
<tr>
<td>Object roles</td>
<td>RO</td>
</tr>
<tr>
<td>Object role dependencies</td>
<td>RD</td>
</tr>
<tr>
<td>Online schema changes</td>
<td>OB</td>
</tr>
<tr>
<td>Package</td>
<td>PG</td>
</tr>
<tr>
<td>Plan</td>
<td>PL</td>
</tr>
<tr>
<td>Procedure</td>
<td>PR</td>
</tr>
<tr>
<td>Routine</td>
<td>FN</td>
</tr>
<tr>
<td>Sequence</td>
<td>SE</td>
</tr>
<tr>
<td>Storage group</td>
<td>SG</td>
</tr>
<tr>
<td>Table</td>
<td>TB</td>
</tr>
<tr>
<td>Trusted contexts</td>
<td>CX</td>
</tr>
<tr>
<td>Trigger</td>
<td>TR</td>
</tr>
<tr>
<td>Table space</td>
<td>TS</td>
</tr>
<tr>
<td>User</td>
<td>US</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 178.

   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

---

**Object type** | **Code**
--- | ---
User name | UN
View | VW
XML relationships | XR
XML strings | XS

*a* Valid if DDF is defined to CATALOG MANAGER.
associated with the object type that was specified in the search. Figure 74 on page 181 shows a search panel for table spaces.

Figure 74: Search panel for table spaces

<table>
<thead>
<tr>
<th>Column</th>
<th>Datatype</th>
<th>Length</th>
<th>Oper</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>VARCHAR</td>
<td>24</td>
<td>=</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>CREATOR</td>
<td>VARCHAR</td>
<td>128</td>
<td>=</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>DBNAME</td>
<td>VARCHAR</td>
<td>24</td>
<td>=</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>OBID</td>
<td>SMALLINT</td>
<td>2</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>OBID</td>
<td>SMALLINT</td>
<td>2</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>PSID</td>
<td>SMALLINT</td>
<td>2</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>BPOOL</td>
<td>CHAR</td>
<td>8</td>
<td>=</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>PARTITIONS</td>
<td>SMALLINT</td>
<td>2</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>LOCKRULE</td>
<td>CHAR</td>
<td>1</td>
<td>=</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>PGSIZE</td>
<td>SMALLINT</td>
<td>2</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>ERASERULE</td>
<td>CHAR</td>
<td>1</td>
<td>=</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>STATUS</td>
<td>CHAR</td>
<td>1</td>
<td>=</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>IMPLICIT</td>
<td>CHAR</td>
<td>1</td>
<td>=</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>NTABLES</td>
<td>SMALLINT</td>
<td>2</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>NACTIVE</td>
<td>INTEGER</td>
<td>4</td>
<td>=</td>
<td></td>
</tr>
</tbody>
</table>

**Note**

The Save/Retrieve Search and Edit a WHERE clause fields are optional. Information is provided in Step 4 on page 182 to Step 7 on page 184.

2 In the Oper column, enter ? to select from a list of operators (only operators that are valid for the value type are displayed) or type the operators to define the type of comparison that CATALOG MANAGER uses for the search.

Table 34 on page 181 lists commonly used operators that are valid on a search panel.

Table 34: Valid search operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Selects objects with values equal to the Value field  &lt;br&gt;This operator is the default for all attributes.</td>
</tr>
<tr>
<td>&lt;</td>
<td>Selects objects with values less than the Value field</td>
</tr>
<tr>
<td>&gt;</td>
<td>Selects objects with values greater than the Value field</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Selects objects that are greater than or equal to the Value field</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Selects objects that are less than or equal to the Value field</td>
</tr>
</tbody>
</table>
Using SEARCH to generate lists based on object attributes

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIKE or L</td>
<td>Selects objects with values that match the wildcard pattern entered in the Value field</td>
</tr>
<tr>
<td>NL</td>
<td>Selects objects with values that do not match the wildcard pattern entered in the Value field</td>
</tr>
<tr>
<td>IN or I</td>
<td>Selects objects which match a list of values in the VALUE field</td>
</tr>
<tr>
<td>NI</td>
<td>Selects objects which do not match a list of values in the VALUE field</td>
</tr>
<tr>
<td>B</td>
<td>Selects objects that are between the two Value fields</td>
</tr>
<tr>
<td>NB</td>
<td>Selects objects that are not between the two Value fields</td>
</tr>
</tbody>
</table>

3 In the Value column, type the values for the fields that CATALOG MANAGER compares 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 (optional) If you want to restrict the search further, create 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:

- The size of the WHERE clause text is limited to 125KB.
- 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.

**Note**
Ensure that the results returned from the query using the WHERE clause are less than 4096KB, as more can cause the search to fail.

You can also use host variables in the WHERE clause of a search statement. For more information, see “Using host variables in a search” on page 185.

5 *(optional)* 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 on page 183) is displayed. Go to Step 6 on page 183.

- Type **R** to retrieve the saved search variables. Press **Enter**.
  The Search Options panel (Figure 76 on page 184) is displayed. Go to Step 7.b on page 184.

6 In the **Save Current Search Variables** section (Figure 75 on page 183), specify values for the search:

**Figure 75: Search Options panel—saving search variables**

| DEEG-R | Save Current Search Criteria | 1 to 10 of 10 | Command ====> | Scroll ====> | PAGE |
|--------|------------------------------|--------------|---------------|-------------|
| Edit before saving | .......................... | N (Y/N) | | | |
| Save using the following identification criteria | Y (Y/N) | | | | |
| For objects of type. | TS | Tablespace | | | |
| Owner. | ........ | MVSJXE1 | | | |
| Name | ........ | | | | |
| Title | ........ | | | | |
| -------------------------- Where clause text to save -------------------------- |
| WHERE A.CREATOR = 'intaqb' FOR FETCH ONLY |

**a** In the **Edit before saving** field, type **Y** or **N** to edit the variables before you save them.

**b** In the **Save using the following identification criteria** field, type **Y**.

CATALOG MANAGER saves the search variables and displays the Search panel. A confirmation message is appended to the panel in the short message field.

**c** 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 “Customizing CATALOG MANAGER command access” on page 329.

To specify another user ID as the search owner, type the appropriate user ID in place of your SQL ID or TSO ID.

d 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.

e (optional) In the Title field, type a description of the search variables in the set. You can use a maximum of 30 characters.

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.

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.

7 In the Retrieve Saved Search Variables section (Figure 76 on page 184), specify identifying values for retrieving saved variables:

**Figure 76: Search Options panel—retrieving search variables**

<table>
<thead>
<tr>
<th>Owner</th>
<th>MVSJXE1</th>
</tr>
</thead>
</table>

a In the Edit after retrieval field, type `Y` or `N` to edit the search variables after the product retrieves them.

b In the Retrieve using the following identification criteria field, type `Y`.

Your user ID is the default value in the Owner field.

c In the Name field, type the name of the set of saved search variables.
You can use wildcard characters that are supported by CATALOG MANAGER in both the Owner and Name fields.

d  Press Enter.

The appropriate search panel is displayed (or a Search Variables List panel if multiple sets of variables match your entry).

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, it is assumed that 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 178.
2 On the Search panel, set the applicable variables.

3 In the **Edit a WHERE clause** field, type **Y**.

4 Create the WHERE clause in the ISPF edit panel.

Define host variables for the **CREATOR** and **NAME** fields as in this example:

**CREATOR LIKE :CREATOR AND NAME = :NAME**

This syntax indicates that the **CREATOR** value can include a wildcard character that is supported by DB2.

5 Press **Enter**.

The SQL Host Variables List panel is displayed.

**Figure 77: SQL Host Variables List panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>SQL Host Variables List</th>
<th>Scrolls</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display SQL . . . : N</td>
<td>(Y/N) Display the SQL statement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute SQL . . . : N</td>
<td>(Y/N) Substitute values and execute SQL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Select a variable with 'S' to enter long character values.

<table>
<thead>
<tr>
<th>S Host Variable</th>
<th>Substitute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATOR</td>
<td></td>
</tr>
<tr>
<td>NAME</td>
<td></td>
</tr>
</tbody>
</table>

6 In the **Substitute Value** column of the **CREATOR** row, enter the value for the **CREATOR** host variable.

**Note**

To enter long host variable values, type **S** by the host variable name for a full-panel edit.

Table 35 on page 186 describes the valid host variable values.

**Table 35: Host variable values**

<table>
<thead>
<tr>
<th>For this value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td><strong>NULL</strong> in uppercase</td>
</tr>
<tr>
<td></td>
<td>To specify <strong>NULL</strong> in the SQL, type <strong>NULL</strong> in uppercase. For all other values, type a number or character value.</td>
</tr>
<tr>
<td>Numeric</td>
<td>Number without quotation marks</td>
</tr>
<tr>
<td></td>
<td>Valid numeric values are not placed in quotation marks.</td>
</tr>
</tbody>
</table>
For this value | Type
---|---
Character | Characters with or without quotation marks
 | 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.

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

8 Choose one of the following actions:

- To see the SELECT statement that CATALOG MANAGER creates, in the **Display SQL** field, type **Y**. Press **Enter**.
  
  The SQL Statement Display panel is displayed. You can only view the SQL statement.
  
- To execute the SELECT statement after specifying the substitute values, in the **Execute SQL** field, type **Y**. Press **Enter**.
  
  CATALOG MANAGER performs the search.
  
  If you type **N** in the **Execute SQL** field, CATALOG MANAGER displays the SQL Host Variables List panel, from which you can continue to make changes to the substitute values. CATALOG MANAGER does not perform the search until you type **Y** in the **Execute SQL** field and press **Enter**.

**Using the Quick-Search feature**

The Quick-Search feature of CATALOG MANAGER enables you to save time by performing searches that use saved search variables or WHERE clauses with fewer steps.

You enter all of the information that CATALOG MANAGER needs to perform the search on the **Command** line of the Primary Menu panel or a list panel. You can enter a maximum of 48 characters.

*Note* 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 205.
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, enter the SEARCH command:

   `SEA objectType searchVariableSet`

   Replace the variables as follows:

   `objectType`

   Represents the two-character code for the object type

   `searchVariableSet`

   Represents the name of the variable set that you want to use

   For example, `SEA DB MMS2.MMSTEST` displays a list that uses the search variables that TSO ID MMS2 saved in MMSTEST.

   **Note**

   If you saved multiple search variable sets under MMSTEST, CATALOG MANAGER would list them so that you could select one. CATALOG MANAGER assumes the current set TSO ID unless you specify another owner. You can also use a wildcard in the entry.

2. Press Enter.

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:

   `SEA objectType abbreviatedWHERE`

   Replace the variables as follows:

   `objectType`

   Represents the two-character code for the object type.
**abbreviated WHERE**

Omit the WHERE operator, and use only correct SQL case, punctuation, syntax, and wildcards that are acceptable to DB2.

For example, `SEA DB NAME LIKE 'DEMO%'` displays a list of databases with names that begin with DEMO.

---

**Creating complex searches**

With CATALOG MANAGER, you can perform, name, save, and retrieve complex searches, such as searches that include JOIN sub-queries.

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 178.
   
   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:

To create a search that contains a JOIN

1. Perform a search for an object.

   For more information, see “Using SEARCH to generate lists based on object attributes” on page 178.

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.

   An example follows:

   , SYSIBM.SYSTABLES B

   c. Type the WHERE operator and WHERE clause. For example:

   , 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')
Note

Remember that you must use correct SQL case, punctuation, syntax, and wildcards that are acceptable to DB2 whenever you use the WHERE clause. For the default object, use a correlation identifier of A.

Customizing object list displays

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 45.
2 On the **Command** line, type **ORDER**.

3 Press **Enter**.

**Figure 78 on page 192** is displayed, listing the columns that are available for the table spaces.

**Figure 78: Column Order Specification panel**

<table>
<thead>
<tr>
<th>Order</th>
<th>Colno</th>
<th>Column Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>TRADITIONAL LIST LINE FORMAT</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>NAME</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>CREATOR</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>DBNAME</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>DBID</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>OBID</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>PSID</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>BPPOOL</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>PARTITIONS</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>LOCKRULE</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>PGSIZE</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>ERASERULE</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>STATUS</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>IMPLICIT</td>
</tr>
<tr>
<td>15</td>
<td>14</td>
<td>NTABLES</td>
</tr>
<tr>
<td>16</td>
<td>15</td>
<td>NACTIVE</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>DSETPASS</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 191), you can display the columns in the traditional list line format again by performing the following steps.

**To display the columns in the traditional list line format**

1. Generate a list of objects.
   
   For information, see “Generating lists in CATALOG MANAGER” on page 45.

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 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.

**To sort a list by one or more columns**

1. Generate a list of table spaces.
For information, see “Generating lists in CATALOG MANAGER” on page 45.

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: Sort Specifications panel**

```
DEFF-R------------------------  SORT SPECIFICATIONS  -------------- ROW 1 OF 46  
Command ===>

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.  
SEQUENCE ASC/DESC NAME  
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v---- 
4        A    NAME  
3        A    CREATOR  
1        A    DBNAME  
  A    DBID  
  A    OBID  
  A    PSID  
  A    BPOOL  
2        D    PARTITIONS  
  A    LOCKRULE  
  A    PGSIZE  
  A    ERASERULE  
  A    STATUS  
  A    IMPLICIT  
  A    NTABLES  
  A    NACTIVE  
  A    DSETPASS  
  A    CLOSERULE
```

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.

**String value search**

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 519.

Use the FIND or RFIND commands to search for a string value in the rows that CATALOG MANAGER displays in the object list.

Counting items

The COUNT command is valid with table spaces, tables, views, aliases, and synonyms. It displays the number of rows in the source objects.

1. Enter COUNT in the Cmd (C) column of a list panel beside a valid object type.

Figure 80: Table Count List panel

<table>
<thead>
<tr>
<th>Qualifier: TABLESPACE=QZUDRR.QZUS34RR</th>
<th>Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZU.QZUT01_DRRS34</td>
<td>5000</td>
</tr>
<tr>
<td>QZU.QZUT02_DRRS34</td>
<td>5000</td>
</tr>
<tr>
<td>QZU.QZUT03_DRRS34</td>
<td>5000</td>
</tr>
<tr>
<td>QZU.QZUT04_DRRS34</td>
<td>5000</td>
</tr>
<tr>
<td>QZU.QZUT05_DRRS34</td>
<td>5000</td>
</tr>
<tr>
<td>QZU.QZUT06_DRRS34</td>
<td>5000</td>
</tr>
<tr>
<td>QZU.QZUT07_DRRS34</td>
<td>5000</td>
</tr>
</tbody>
</table>

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: DB2 Catalog Counts panel**

![DB2 Catalog Counts panel](image)

<table>
<thead>
<tr>
<th>Object Type</th>
<th>Row Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>538</td>
</tr>
<tr>
<td>AuxRels</td>
<td>11809</td>
</tr>
<tr>
<td>Database</td>
<td>3124</td>
</tr>
<tr>
<td>Datatypes</td>
<td>855</td>
</tr>
<tr>
<td>Collection</td>
<td>993</td>
</tr>
<tr>
<td>Columns</td>
<td>720625</td>
</tr>
<tr>
<td>DBRM</td>
<td>4962</td>
</tr>
<tr>
<td>Indexes</td>
<td>53465</td>
</tr>
<tr>
<td>Package</td>
<td>38042</td>
</tr>
<tr>
<td>Plan</td>
<td>2939</td>
</tr>
<tr>
<td>Routines</td>
<td>2920</td>
</tr>
<tr>
<td>Stogroup</td>
<td>542</td>
</tr>
<tr>
<td>Strings</td>
<td>1380</td>
</tr>
<tr>
<td>Synonyms</td>
<td>11670</td>
</tr>
<tr>
<td>Tables</td>
<td>49249</td>
</tr>
<tr>
<td>Tablespace</td>
<td>34884</td>
</tr>
<tr>
<td>Triggers</td>
<td>602</td>
</tr>
<tr>
<td>Userauth</td>
<td>1364</td>
</tr>
<tr>
<td>Views</td>
<td>8789</td>
</tr>
</tbody>
</table>

You can use CATALOG MANAGER commands to display object descriptions.

The following commands are available:

- **DESCRIBE**
- **DES**
- **D**
- **S**
- **DESTATISTICS**

Each command meets a specific need. You can print the descriptive information if a hard copy 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 on page 198 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 93.)

**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: QZU.QZUT01_DEBS01
Command ===>                                                  Scroll ===> PAGE
Table = QZU.QZUT01_DEBS01                                    
------------------------------------------------------------------------------
| Creator  | QZU  | Type  | T |
| Database  | QZUDEB | Status | X |
| Tablespace | QZUS01EB | Checkflag | .X'40404040' |
| Editproc |  | Validproc | . |
| Audit |  | Parents | 0 |
| Pctpages | 75 |  | |
| Colcount | 12 | DBID | .1204 |
| Record Length | 86 | OBID | . |
| Key Columns | 5 | Key OBID | X'4040404040' |
| Createdby | RDABKH1 | Label | . |
| Alteredts | ..-06-10.40.20.849236 | Datacapture | . |
| Rba1 | X'006009FB9000' | Rba2 | X'006009FB9000' |
| Pctrowcomp | 70 |  | |
| Orphan |  | Clustertype | . |
| IBM reqd | I | Checks | 0 |
| Viewdeps |  | Cardf | 988013 |
| Checkrid5b | X'4040404040' | Encoding scheme | .E |
| Tbcreator |  | Tbname | . |
| Npagesf | 14768 | Spacef | 79920 |
| Avgrowlen | 60 |  | |
| Tablestatus | 0 | Number dep MQTs | 0 |
| Version |  | Partkeycolnum | 0 |
| Split Rows |  | Security Label | . |
| Owner | QZU | Append | N |
| Ownertype |  |  |  |
|------------------------------------------------------------------------------
| TABLE CONSTRAINTS
<table>
<thead>
<tr>
<th>Constraint Name</th>
<th>Type</th>
<th>Enforcing Index</th>
<th>Creator</th>
<th>Colcount</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>P</td>
<td>QZU.QZUX02_DEBS01T01</td>
<td>RDABKH1</td>
<td>5</td>
</tr>
<tr>
<td>-----------------</td>
<td>------</td>
<td>------------------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>TABLE COLUMNS</td>
<td>Num Column Name</td>
<td>Coltype</td>
<td>Length</td>
<td>Ni</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------</td>
<td>---------</td>
<td>--------</td>
<td>----</td>
</tr>
<tr>
<td>1 DATE</td>
<td>DATE</td>
<td>4</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>2 AUTHID</td>
<td>CHAR</td>
<td>8</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>3 TBCREATOR</td>
<td>CHAR</td>
<td>8</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>4 TBNAME</td>
<td>CHAR</td>
<td>18</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>5 PLAN</td>
<td>CHAR</td>
<td>8</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>6 LAST_MNT_DATE_TIME</td>
<td>TIMESTAMP</td>
<td>10</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>7 NTRANS</td>
<td>INTEGER</td>
<td>4</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>8 INSERTS</td>
<td>INTEGER</td>
<td>4</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>9 UPDATES</td>
<td>INTEGER</td>
<td>4</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>10 DELETES</td>
<td>INTEGER</td>
<td>4</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>11 MIN_TIME</td>
<td>TIME</td>
<td>3</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>12 MAX_TIME</td>
<td>TIME</td>
<td>3</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------</td>
<td>---------</td>
<td>--------</td>
<td>----</td>
</tr>
<tr>
<td>Row Length - maximum 86 - minimum 86 including eight byte header</td>
<td>END OF DATA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**DES command**

The DES command, for some object types, displays a subset of the description that is provided by the DESCRIBE command.
The affected object types and descriptive information are shown in Table 36 on page 199.

Table 36: DES command descriptions

<table>
<thead>
<tr>
<th>Object types</th>
<th>Excluded information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Indexes, foreign keys, table partition columns, table partition limit keys, and materialized query table (MQT) text</td>
</tr>
<tr>
<td>Database</td>
<td>Plans</td>
</tr>
<tr>
<td>DBRM package</td>
<td>Explain access information from the PLAN_TABLE</td>
</tr>
</tbody>
</table>

For object types that are not shown in Table 36 on page 199, 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 on page 200 shows the Describe Table panel generated from the
DESTATISTICS command for the example table that is used in “DESCRIBE
command” on page 197.

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.
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 33).

To write the contents of an entire list to your print data set

1  Enter PRI on the Command line of the list panel.

To generate a DESCRIBE report and write it to your print data set

1  Enter PRI in the Cmd column of a list.

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

You can run a batch command in a DB2 object list or mixed list.

1  Generate a list:

   ■ To generate an object list, see “Generating lists in CATALOG MANAGER” on page 45.

   ■ To generate a mixed list, see “Generating a mixed list” on page 174.

2  If you generated an object list, type command BATCH, DSNcommand BATCH or DB2command BATCH in the Cmd column of the source object, and press Enter.
You can use any of the following commands:

<table>
<thead>
<tr>
<th>command</th>
<th>DSNcommand</th>
<th>DB2command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASCADE</td>
<td>BIND</td>
<td>DISPLAY</td>
</tr>
<tr>
<td>CONNECT (can only be issued from a location, or LO, list)</td>
<td>FREE</td>
<td>START</td>
</tr>
<tr>
<td>DCL</td>
<td>REBIND</td>
<td>STOP</td>
</tr>
<tr>
<td>DDL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCRIBE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESTATISTICS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HDDL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HGRANT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CATALOG MANAGER displays the following message:

BMC14651 Use BATCH command to generate JCL for all saved commands.

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

The CATALOG MANAGER Batch Job panel is displayed.

**Figure 84: CATALOG MANAGER Batch Job panel**

DEFF-------------------------  CATALOG MANAGER Batch Job  ----------
Command ===>                     CATALOG MANAGER Batch Job  ----------
JCL Dataset . . . 'RDACRJ.DBDC.JCL(BATCH)'
Set JCL options . N (Y/N - Change options for creating JCL)
Build Job . . . . Y (Y/N - Create JCL, save in JCL dataset)
Edit Dataset . . . Y (Y/N - Edit JCL dataset)
Submit . . . . . . N (Y/N - Submit JCL dataset)
--------------------------------  Job Options  -------------------------------
Ssid . . . . . . . DEDK The ssid of the DB2 which batch is to connect to
Default options . DC910EDK The name of the default options to use in batch
HDDL output dsn &ZUSER..&SSID..HDDL

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(0=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 “Connecting to a specified SSID” on page 161.

**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 on page 204 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><code>DISPLAY objectType objectName</code></td>
<td>DB, TS</td>
<td>DB, TS, FN P 12.a , PR 12.a</td>
</tr>
<tr>
<td><code>START objectType objectName</code></td>
<td>DB, TS, FN P 12.a P 12.b , PR 12.a P 12.b</td>
<td></td>
</tr>
<tr>
<td><code>STOP objectType objectName</code></td>
<td>DB, TS, FN P 12.a P 12.b , PR 12.a P 12.b</td>
<td></td>
</tr>
</tbody>
</table>

a Part 1 of `objectName` cannot exceed 8 characters, and part 2 cannot exceed 18 characters.

b 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 0 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

You can use a batch command in a CATALOG MANAGER list or search.

1. On the Command line of a Primary Menu panel or a list panel, type BATCH and press Enter.

   The CATALOG MANAGER Batch Job panel is displayed.

2. In the Build Job field, type Y to create the JCL.

3. Near the bottom of the JCL, find the NO CATALOG MANAGER COMMANDS message in the SYSIN stream (Figure 85 on page 205).

4. 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.

5 Press END to return to the CATALOG MANAGER Batch Job panel.

6 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.

For more information, view the Quick Course "Managing SQL."

**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 on page 207 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 86).

**Figure 86: Confirm SQL panel for CREATE table**

```
DEFF-R                      Confirm SQL                         1 to 11 of 11
Command ===>                                                  Scroll ===> CSR
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) . . . . . . . .  ACT.V10.DATABASE(TEST)
Execute SQL . . . . . . . . .  N       Y/N Execute the SQL
-------------------------------------  SQL  -----------------------------------
CREATE TABLE
RDACRJ.T0027_CLAIM
(  CLAIM_NUM DECIMAL(10) NOT NULL WITH DEFAULT
 ,ACTIVITY_TYPE_CD CHAR(3) NOT NULL WITH DEFAULT
 ,CLAIM_OVP_AMT DECIMAL(11) NOT NULL WITH DEFAULT
 ,CLAIM_STS_CD CHAR(2) NOT NULL WITH DEFAULT
 ,CREATION_DT DATE NOT NULL WITH DEFAULT
 )
IN DEMOCRJ.CLAIMTS
;
```

**Using 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.

To display the SQL_Table List that includes only SQL of which you are the owner

1. Perform one of the following tasks:

   - Type SQL on the Command line of the Primary Menu panel or a list panel, and then press Enter.
- Type SQL *.* on the Command line of the Primary Menu Panel or a list panel, and then press Enter.

Figure 87 on page 208 shows a section of a sample SQL_Table List panel.

Figure 87: 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.</td>
<td>CSR</td>
</tr>
<tr>
<td>Subcommands are: ANALYZE, CUT, DELETE, EDIT, EXECUTE, PASTE, RENAME, TBBROWSE, TBEDIT, 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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Owner</th>
<th>Name</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDAMSL</td>
<td>PBRTSTB</td>
<td>CREATE TABLESPACE MSLTSPBR IN MSLTMPDB LOGGE</td>
<td></td>
</tr>
<tr>
<td>RDAPKM</td>
<td>20100317_111620</td>
<td>DSN SYSTEM(DEFF) DCLGEN TABLE(SYSTABLESPACES</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>CREATE_TB</td>
<td>CREATE TABLE PXB.CURRENTQUOTE ( SYMBOL CHAR(</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>CX</td>
<td>CREATE TRUSTED CONTEXT QCH_TRUSTD_CONTEXT BA</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>DROP_INCLUDE</td>
<td>CREATE UNIQUE INDEX TAD.TAD_TBL01_IX ON TAD.</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>DTS</td>
<td>CREATE DISTINCT TYPE QCH.QCH_IDENT_1 AS INTE</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>FN</td>
<td>CREATE FUNCTION QCH.QCH_FN_EXT_TYPE_T ( LAST</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>HDDL</td>
<td>-- HDDL OBJECTS PRODUCED FOR: TS TB VW IX SY</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>PR</td>
<td>CREATE PROCEDURE QCH.QCHCT007 ( IN INAME VAR</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>SYSTEM_DBA</td>
<td>GRANT DBADM WITHOUT ACCESSCTRL WITHOUT DATAA</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>TEST</td>
<td>-SQLP 000100 CREATE PROCEDURE TAD.TAD_NATIVE</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>TEST-109</td>
<td>-- HDDL OBJECTS PRODUCED FOR: TS TB VW IX SY</td>
<td></td>
</tr>
<tr>
<td>RDAPXB</td>
<td>TRUNCATE_TEST</td>
<td>-- HDDL OBJECTS PRODUCED FOR: TS TB VW IX SY</td>
<td></td>
</tr>
<tr>
<td>RDAPXB2</td>
<td>EXECUTION</td>
<td>-- HDDL OBJECTS PRODUCED FOR: TS TB VW IX SY</td>
<td></td>
</tr>
</tbody>
</table>

Copying SQL in the SQL_Table

Use the following procedure 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 207.

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 74.

4. Edit the SQL as necessary.

5. Press END to save the SQL.

Copying external SQL

Use the following procedure to copy SQL that has a similar LRECL from a PDS or sequential file from a source that is outside the SQL_Table.
1 From the Primary Menu panel, type SQL EDIT newName on the Command line.

2 Press Enter.

   An ISPF edit panel is displayed.

3 On the Command line, type COPY.

4 Press Enter.

   The ISPF Edit/View-Copy panel is displayed.

5 Specify the library member or sequential data set that you want to copy.

6 Press Enter.

   The member or sequential data set is displayed in the ISPF edit panel with Member member-name copied in the short message area of the panel.

7 Edit the member, or press END to save the SQL.

Applying SQL model statements

From any list panel, you can apply an SQL model statement that contains host variables to the list objects by using the APPLY command.

For the host variables in the SQL model statement, CATALOG MANAGER substitutes values in the DB2 catalog row indicated by the list entry and creates one statement for each object in the list. The host variables must be DB2 catalog column names.

By using the APPLY command, you can significantly reduce the time required to perform the same action against a group of objects. For example, using one model statement, you can add the RESTRICT ON DROP attribute to a group of tables.

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.

To apply SQL statements to lists

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 45.

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: Confirm APPLY SQL MODEL panel**

   ![Confirm APPLY SQL MODEL panel](image)

   DEFF-R ------------------- Confirm APPLY SQL MODEL -------------------
   Command ===> Edit SQL Model . . . . . N  (Y/N)  Current SQLID : RDAJR2
   Replace with SQL Member .
   Save model in SQL table . N  (Y/N)
   Apply to list objects . . N  (Y/N)
   Name of saved SQL . . . . . RDAJR2.RESTRICT
   ------------------------- SQL -----------------------------------
   More:  +
   ALTER TABLE :CREATOR . :NAME ADD RESTRICT ON DROP;
   ******************************** BOTTOM OF DATA ****************************

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**.

210  *CATALOG MANAGER for DB2 User Guide*
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 on page 211). Values from the DB2 catalog have been substituted for the host variables.

**Figure 89: Confirm SQL panel**

```
DEFF-R --------------------------  Confirm SQL  ------------ Row 1 to 15 of 235
Command ===>                                                  Scroll ===> PAGE 02
Actions  Edit options . . . .  N  (Y/N)                      
          Edit SQL . . . . . .  N  (Y/N)  Current SQLID : RDACRJ2
          Save SQL . . . . . .  N  (A/R/Y/N A-append, R-replace, Y-append)
          Execute . . . . . .  N  (Y/N)                      
Options  Name of saved SQL     20110209_325680
          ------------------------------------  SQL  ------------------------------------
          ALTER TABLE  QZU    .   QZUT01_DCI08S01
          ALTER TABLE  QZU    .   QZUT02_DCI08S01
          ALTER TABLE  QZU    .   QZUT03_DCI08S01
```

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.

d *(optional)* In the **Save SQL** field, type **Y** to save the SQL in the CATALOG MANAGER SQL_Table. In the **Name of saved SQL** field, type a name for the SQL. Then, press **Enter**.
The saved SQL uses the ID displayed in the **Current SQLID** field as the object qualifier. If the SQL is not saved, the ID in the **Current SQLID** is used only to identify DB2 authority.

**e (optional)** In the **Execute** field, type **Y** to execute the SQL displayed on the Confirm SQL panel. Then, press **Enter**.

The SQL Progress Indicator panel is displayed. The panel automatically refreshes to display the status of the SQL that is being executed.

---

### Using 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 45.
   - 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: Confirm SQL panel for extended SQL processing**

```
<table>
<thead>
<tr>
<th>Command ==&gt;</th>
<th>Scroll ==&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC14601</td>
<td>Changes were made during the EDIT call</td>
</tr>
<tr>
<td>Current SQLID:</td>
<td>RDACRJ2</td>
</tr>
<tr>
<td>Edit Options:</td>
<td>Y/N Modify SQL processing options</td>
</tr>
<tr>
<td>Edit SQL:</td>
<td>Y/N Edit SQL before executing</td>
</tr>
<tr>
<td>Save in SQL table:</td>
<td>A/Y/R/N A/Y-Append, R-Replace</td>
</tr>
<tr>
<td>Name of saved SQL:</td>
<td>20110121_115552</td>
</tr>
<tr>
<td>Save in PDS:</td>
<td>Y/N Save SQL in PDS</td>
</tr>
<tr>
<td>PDS(member):</td>
<td>ACT.V10.DATABASE(TEST)</td>
</tr>
<tr>
<td>Analysis:</td>
<td>Y/N Call SQL Explorer for EXPLAIN</td>
</tr>
<tr>
<td>Edit/Browse data:</td>
<td>E/B/N Call the Table Editor</td>
</tr>
<tr>
<td>Execute SQL:</td>
<td>Y/N Execute the SQL</td>
</tr>
</tbody>
</table>
```

```
SELECT
COLUMN_1                        ,-- =                    --DATE
COLUMN_2                        ,-- =                    --SMALLINT
COLUMN_3                        ,-- ='            '      --CHAR(12)
COLUMN_4                        ,-- ='       '           --CHAR(7)
COLUMN_5                        ,-- ='    '              --CHAR(4)
FROM ACT.ACTC01_D92S01
WHERE COLUMN_1 = :DATE AND COLUMN_3 = :FUNCTION
```

3 In the **Execute SQL** field, type **Y**.

4 Press **Enter**.

The SQL Host Variables List panel is displayed.

**Figure 91: SQL Host Variables List panel**

```
<table>
<thead>
<tr>
<th>Command ==&gt;</th>
<th>Scroll ==&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display SQL:</td>
<td>N</td>
</tr>
<tr>
<td>Execute SQL:</td>
<td>N</td>
</tr>
</tbody>
</table>
```

Select a variable with 'S' to enter long character values.

```
S Host Variable | Substitute Value
-----------------|-----------------|
DATE            | FUNCTION
```

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

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

7 Choose one of the following actions:
To see the SELECT statement that CATALOG MANAGER creates, in the Display SQL field, type Y. Press Enter.

The SQL Statement Display panel is displayed. You can only view the SQL statement.

To execute the SELECT statement after specifying the substitute values, in the Execute SQL field, type Y. Press Enter.

CATALOG MANAGER performs the search.

If you type N in the Execute SQL field, CATALOG MANAGER displays the SQL Host Variables List panel, from which you can continue to make changes to the substitute values. CATALOG MANAGER does not perform the search until you type Y in the Execute SQL field and press Enter.

Where to go from here

An important feature of CATALOG MANAGER is the ability to view and edit the data in the catalog tables without exiting the product.

“Browsing and editing data” on page 215 explains how to invoke the data editing and browsing function after you have created lists of the tables or views with which you want to work.
Browsing and editing data

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 “Commands” on page 501.

For more information, view the Quick Course "Using the Table Editor."

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.
Related Information

- “Commands” on page 501

Methods for invoking the data browsing function

The following table lists the available methods for invoking the data browsing function.

Table 39: Commands to invoke data browsing

<table>
<thead>
<tr>
<th>Data browsing command</th>
<th>Where to enter command</th>
</tr>
</thead>
<tbody>
<tr>
<td>BROWSE TB <code>owner. tablename</code></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><strong>Edit/Browse</strong> field on Confirm SQL panel for editing <strong>SELECT SQL</strong> 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

Follow this procedure to set the options of the data browsing function.

1. Invoke the data browsing function.

For more information, see “Methods for invoking the data browsing function” on page 216.
The Browse DB2 Table Options panel is displayed.

**Figure 92: Browse DB2 Table Options panel**

<table>
<thead>
<tr>
<th>DEFF-R</th>
<th>Browse DB2 Table Options</th>
<th>Command =&gt;</th>
</tr>
</thead>
</table>

Specify the following options. Press ENTER to continue or END to exit.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current SQLID</td>
<td>RDACRJ</td>
</tr>
<tr>
<td>Current Authid</td>
<td>RDACRJ</td>
</tr>
<tr>
<td>Table name or pattern</td>
<td>QZU.QZUT00_DSC30S28</td>
</tr>
<tr>
<td>Edit select statement</td>
<td>N (Display and edit select statement.)</td>
</tr>
<tr>
<td>Save/Retrieve Select</td>
<td>N (S/R/N) N-no action</td>
</tr>
<tr>
<td>Select statement name</td>
<td>. . .</td>
</tr>
<tr>
<td>Display selected row count</td>
<td>N (Display count of rows matching WHERE clause.)</td>
</tr>
<tr>
<td>Select row limit</td>
<td>300 (0-99999999 rows to edit. 0 = no limit.)</td>
</tr>
<tr>
<td>Initial Display Mode</td>
<td>C (C-column view, R-row view)</td>
</tr>
<tr>
<td>Display additional options</td>
<td>N (Display panel to specify browsing options.)</td>
</tr>
</tbody>
</table>

2. In the **Current SQLID** field, type a different SQLID.

3. In the **Table name or pattern** field, type the name of a table or type a pattern.

   **Note**

   Wildcards are valid in the **Table name or pattern** field. For information on wildcards, see “Supported wildcards in qualifiers” on page 40.

4. In the **Edit select statement** field, type **Y** to display or edit the SELECT statement that CATALOG MANAGER generates.

The Select Statement Specification panel is displayed.

**Figure 93: Select Statement Specification panel**

<table>
<thead>
<tr>
<th>DEFF</th>
<th>Select Statement Specification</th>
<th>Command =&gt;</th>
<th>Scroll =&gt; CSR</th>
</tr>
</thead>
</table>

Select columns to edit and type where clause values, then press ENTER

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Length</th>
<th>Order</th>
<th>OPER</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_1</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_2</td>
<td>SMALLINT</td>
<td>2</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_3</td>
<td>CHAR</td>
<td>1</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_4</td>
<td>VARCHAR</td>
<td>1</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_5</td>
<td>DECIMAL</td>
<td>5,2</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_6</td>
<td>VARCHAR</td>
<td>1</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_7</td>
<td>CHAR</td>
<td>3</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_8</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_9</td>
<td>FLOAT</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_10</td>
<td>VARCHAR</td>
<td>1</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_11</td>
<td>FLOAT</td>
<td>8</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_12</td>
<td>DATE</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_13</td>
<td>CHAR</td>
<td>1</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_14</td>
<td>TIME</td>
<td>3</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_15</td>
<td>VARCHAR</td>
<td>1</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_16</td>
<td>TIMESTMP</td>
<td>10</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>COLUMN_17</td>
<td>VARCHAR</td>
<td>1</td>
<td>A</td>
<td>=</td>
<td></td>
</tr>
</tbody>
</table>

5. **(optional)** If needed customize the SELECT statement.
a Make the specifications that are shown in Table 40 on page 218 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 5.a.1</td>
<td>Delete the S designator beside the column name.</td>
</tr>
<tr>
<td>Specify a sequence of columns 5.a.1</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 “Using SEARCH to generate lists based on object attributes” on page 178.</td>
</tr>
<tr>
<td>Specify values for the search operation 5a.ii</td>
<td>Type the values in the Value field.</td>
</tr>
</tbody>
</table>

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

b If you specify a search value for a column in the Value field, then the applicable column will be included in the SELECT statement even if you do not select the column with the S designator or a sequence number.

b After customizing the SELECT statement, press Enter to display the panel with your specifications.

c (optional) Use the command-line commands that are shown in Table 41 on page 218 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>

d Press END to display the Browse DB2 Table Options panel.

6 In the Save/Retrieve Select field, type S, R, or N to indicate your action on the SELECT statement:
(optional) Use the command-line commands that are shown in Table 41 on page 218 to clear changes that you have made.

Table 42: 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>

<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>

7 If you typed R in the Save/Retrieve Select field in Step Setting options for browsing data on page 218, 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.

8 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.

9 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.

10 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.

11 In the Display additional options field, type Y or N to display the Edit and Browse Options panel, from which you can specify additional values.

**Figure 94: Edit and Browse Options panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Edit and Browse Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>More: +</td>
<td></td>
</tr>
</tbody>
</table>

Specify the following options, then press END to exit.

- **Display Headings** . . . . . . 1 1-Names, 2-Labels, 3-Numbers
- **Date and Time defaults** . . . 2 Select value for columns with no DB2 default
  - 1 - 1900-01-01-00.00.00.000000
  - 2 - Current date and time
  - 3 - Blank
- **Display update SQL** . . . . . N Display SQL used to update the table.
- **Confirm before update** . . . N Display confirm panel before saving changes.
- **Display Statistics** . . . . . Y Display Table Editing Statistics.
- **Browse with UR** . . . . . . . Y Browse DB2 rows using WITH UR in select.
- **CAPS ON** . . . . . . . . . . N Uppercase input and modified values.
- **Default SQL Owner** . . . . . S Default owner for SQL unqualified sql table member names when saving or retrieving SQL.
  - T - TSO ID
  - S - SQL ID
- **Memory Allocation Limit** . . O Meg. Maximum memory to allocate for DB2 rows, in megabytes. 0 = No limit.
- **Left Justify Numerics** . . . N Numeric fields in rowview mode will be displayed left justified.
- **Display Decimal Point** . . . N Include a decimal point in the value of columns with type DECIMAL(n,0)
- **Clear Editor Users Table** . . N (Y/N) A SYSADM may need to delete ALL rows from the editor users table, where Catalog Manager records who is editing which tables.

**Table 43 on page 220** shows the options that you can specify on the Edit and Browse Options panel.

**Table 43: 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>
</tbody>
</table>
### Browsing data

Use the following procedure to browse data.

1. Invoke the data browsing function. For more information, see “Methods for invoking the data browsing function” on page 216.

2. Set your options for browsing data. For more information, see “Setting options for browsing data” on page 216.

3. Press Enter.

   The Browse DB2 Table panel is displayed.

4. After you have browsed the data, press END.
Browsing data in LOB columns

Use the following procedure to browse data in a LOB column.

**Note**

You cannot use the data editing function to edit data in 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 “Methods for invoking the data browsing function” on page 216.

3. In the Browse DB2 Table Options panel, specify your options.

   For more information, see “Setting options for browsing data” on page 216.

4. In the Browse DB2 Table panel, perform the following steps to view the entire value for a LOB column:

   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.
Methods for invoking the data editing function

The following table lists the available methods for invoking the data editing function.

Table 44: Commands to invoke data editing

<table>
<thead>
<tr>
<th>Data editing command</th>
<th>Where to enter command</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDIT TB owner.tableName</td>
<td>Command line of Primary Menu panel or list panel</td>
</tr>
<tr>
<td>EDIT ED</td>
<td>Cmd (C) column of source table on Table List panel</td>
</tr>
<tr>
<td>E</td>
<td>Edit/Browse field on Confirm SQL panel for editing SELECT SQL statement</td>
</tr>
<tr>
<td>TBEDIT</td>
<td>SELECT SQL statement line of SQL_Table List panel</td>
</tr>
</tbody>
</table>

Setting options for editing data

Use the following procedure to set the editing options.

1. Invoke the data editing function.

   For more information, see “Browsing table data” on page 215.

   The Edit DB2 Table Options panel is displayed (Figure 95 on page 224). The Edit DB2 Table Options panel provides the same options as the Browse DB2 Table Options panel (see “Setting options for browsing data” on page 216), with the following additions:

   - Hold rows during edit
   - Edit or Browse mode
Figure 95: Edit DB2 Table Options panel

Specify the following options. Press ENTER to continue or END to exit.

<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current SQLID</td>
<td>RDACRJ2</td>
</tr>
<tr>
<td>Current Authid</td>
<td>RDACRJ</td>
</tr>
<tr>
<td>Table name or pattern</td>
<td>QZU.QZUT00_DSC30S2B</td>
</tr>
<tr>
<td>Edit select statement</td>
<td>N</td>
</tr>
<tr>
<td>Save/Retrieve Select</td>
<td>N</td>
</tr>
<tr>
<td>Display select statement</td>
<td>N</td>
</tr>
<tr>
<td>S-save current select statement in SQL table using the select statement name matching pattern in select statement name</td>
<td></td>
</tr>
<tr>
<td>R-retrieve list of saved select statements matching pattern in select statement name</td>
<td></td>
</tr>
<tr>
<td>Select statement name</td>
<td></td>
</tr>
<tr>
<td>Display selected row count</td>
<td>N</td>
</tr>
<tr>
<td>Select row limit</td>
<td>300</td>
</tr>
<tr>
<td>Hold rows during edit</td>
<td>R</td>
</tr>
<tr>
<td>Initial Display Mode</td>
<td>C</td>
</tr>
<tr>
<td>Edit or Browse mode</td>
<td>E</td>
</tr>
<tr>
<td>Copy Table Rows</td>
<td>N</td>
</tr>
<tr>
<td>Display additional options</td>
<td>N</td>
</tr>
<tr>
<td>Display panel to specify editor options</td>
<td></td>
</tr>
</tbody>
</table>

2 Follow the steps in “Setting options for browsing data” on page 216.

3 In the Hold rows during edit field, type T, R, or N to indicate how requests for edits from other users are handled while you are editing data:

<table>
<thead>
<tr>
<th>To specify this locking option</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shared table lock</td>
<td>T</td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Row lock</td>
<td>R</td>
</tr>
<tr>
<td>CATALOG MANAGER does not issue any locks and allows DB2 to perform normal lock escalation. If the table space in which you are editing was created with LOCKSIZE ROW, row locks can be used unless DB2 performs lock escalation based on the number of rows edited. Any changes that you make are committed when you exit the editing function to the Edit DB2 Table Options panel.</td>
<td></td>
</tr>
</tbody>
</table>
To specify this locking option | Type
---|---
**No lock**<br>CATALOG MANAGER does not issue any locks and allows DB2 to perform normal lock escalation.<br>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:<br>■ If no changes have occurred, the edits are applied and committed.<br>■ 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.

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 “How CATALOG MANAGER handles lock contention” on page 226.

**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.

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 (see “Methods for invoking the data editing function” on page 223).
Note
You can invoke data editing by issuing one of the data editing commands (see "Methods for invoking the data editing function" on page 223), and then switch to data browsing by setting the **Edit or Browse mode** value to B. In this situation, you can switch back to data editing by resetting the **Edit or Browse mode** value to E.

5 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.

---

**How CATALOG MANAGER handles 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 Reference Manual*, *Installation System Quick Start*, the *BMC Products and Solutions for DB2 Customization Guide*, and the IBM documentation.

---

**Editing data**

Use the following procedure to edit data.

1 Invoke the data editing function. For more information, see “Methods for invoking the data browsing function” on page 216.
2 Set the options for controlling the display and the SQL processing. For more information, see “Setting options for editing data” on page 223.

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 on page 227) according to the Initial Display Mode value that was selected in the Edit DB2 Table Options panel (see “Setting options for editing data” on page 223).

**Figure 96: Edit DB2 Table panel in column view mode**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF----------</td>
<td>Edit DB2 Table --------</td>
</tr>
</tbody>
</table>

QZU.QZUT01_DCI19S05 (1/300)

<table>
<thead>
<tr>
<th>000001</th>
<th>1818599</th>
<th>0</th>
<th>AABN0000467</th>
<th>1124064</th>
<th>4064</th>
<th>'DEFENDANT ARRAIGN'</th>
</tr>
</thead>
<tbody>
<tr>
<td>000002</td>
<td>1818629</td>
<td>7364</td>
<td>AABN0000478</td>
<td>1130064</td>
<td>0064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000003</td>
<td>1818687</td>
<td>26700</td>
<td>AABN0000495</td>
<td>1139064</td>
<td>9064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000004</td>
<td>1818706</td>
<td>-20104</td>
<td>AABN0000501</td>
<td>1141064</td>
<td>1064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000005</td>
<td>1818722</td>
<td>27713</td>
<td>AABN0000506</td>
<td>1145064</td>
<td>5064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000006</td>
<td>1818733</td>
<td>-14913</td>
<td>AABN0000510</td>
<td>1146063</td>
<td>6063</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000007</td>
<td>1818754</td>
<td>7200</td>
<td>AABN0000517</td>
<td>1148064</td>
<td>8064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000008</td>
<td>1818758</td>
<td>16000</td>
<td>AABN0000519</td>
<td>1148064</td>
<td>8064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000009</td>
<td>1818781</td>
<td>7300</td>
<td>AABN0000525</td>
<td>1160064</td>
<td>0064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000010</td>
<td>1818810</td>
<td>3300</td>
<td>AABN0000536</td>
<td>1164064</td>
<td>4064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000011</td>
<td>1818836</td>
<td>6800</td>
<td>AABN0000543</td>
<td>1171064</td>
<td>1064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000012</td>
<td>1818895</td>
<td>0</td>
<td>AABN0000561</td>
<td>1176064</td>
<td>6064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000013</td>
<td>1818902</td>
<td>0</td>
<td>AABN0000562</td>
<td>1177064</td>
<td>7064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000014</td>
<td>1818932</td>
<td>4275</td>
<td>AABN0000573</td>
<td>1182064</td>
<td>2064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000015</td>
<td>1818944</td>
<td>0</td>
<td>AABN0000577</td>
<td>1184064</td>
<td>4064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000016</td>
<td>1818954</td>
<td>5985</td>
<td>AABN0000578</td>
<td>1185064</td>
<td>5064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000017</td>
<td>1447973</td>
<td>18000</td>
<td>AABN0000585</td>
<td>1655233</td>
<td>5233</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000018</td>
<td>1819036</td>
<td>0</td>
<td>AABN0000600</td>
<td>1209064</td>
<td>9064</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000019</td>
<td>1448522</td>
<td>0</td>
<td>AABN0000603</td>
<td>0773625</td>
<td>3625</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
<tr>
<td>000020</td>
<td>1448585</td>
<td>-31536</td>
<td>AABN0000612</td>
<td>0774002</td>
<td>4002</td>
<td>'DEFENDANT ARRAIGN'</td>
</tr>
</tbody>
</table>

**Note**

If the selected table contains no data, then the placeholder *INS is displayed.

4 Edit the table using the following commands:

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit data</td>
<td>Move the cursor to the column and row and type the new values. You can switch between column and row views. You can also make changes using the command line: Type C originalString changedString or CHANGE originalString changedString in the Command line.</td>
</tr>
</tbody>
</table>

**Note**: Any changes that you have made to the data are highlighted before the viewing mode is switched.
<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View data</td>
<td>To switch from column view to row view, type <strong>ROW</strong> on the <strong>Command</strong> line, and then press <strong>Enter</strong>. To switch back to column view, click END. To view additional rows of data, press <strong>F11</strong> to go to the next row or <strong>F10</strong> to return to previous row. <strong>Note:</strong> You can change your default view to Row view by changing your table editor options. See “Setting options for editing data” on page 223.</td>
</tr>
<tr>
<td>Insert row</td>
<td>Type <strong>I</strong> next to the row you want to precede the insertion</td>
</tr>
<tr>
<td>Repeat row</td>
<td>Type <strong>R</strong> next to the row that you want to repeat</td>
</tr>
<tr>
<td>Delete row</td>
<td>Type <strong>D</strong> next to the row that you want to delete. *DEL is displayed <strong>Note:</strong> Any deletions are highlighted.</td>
</tr>
<tr>
<td>Zoom row</td>
<td>Type <strong>Z</strong> to zoom the row and enter row view. To update a column with a hex value, zoom the row and enter the update on the zoomed field.</td>
</tr>
<tr>
<td>Undo uncommitted changes</td>
<td>Type <strong>U</strong> and press <strong>Enter</strong> next to the row that you want to undo any uncommitted changes.</td>
</tr>
<tr>
<td>Commit changes</td>
<td>To save updated, inserted, or deleted data, 1. Click END or type <strong>SAVE</strong> on the <strong>Command</strong> line (commit and return you to the table editor) or END (commit and exit the table editor) on the command line. The Statistics panel is displayed. 2. Review and press <strong>Enter</strong> to save or type <strong>CANCEL</strong> to rollback the changes <strong>Note:</strong> To leave the table editor without committing your changes, type <strong>CAN</strong> or <strong>CANCEL</strong> on the <strong>Command</strong> line.</td>
</tr>
<tr>
<td>Sort viewed data</td>
<td>1. Type <strong>SORT columnName order</strong> to change the order of the data that you are viewing. You are prompted to define column sequence and order (ascending/descending). 2. Click END and the table editor displays changes.</td>
</tr>
<tr>
<td>Find data</td>
<td>Type <strong>F</strong> <strong>SearchString</strong> or <strong>FIND SearchString</strong> in the <strong>Command</strong> line</td>
</tr>
</tbody>
</table>

**Note**
The online Help (click **F1**) contains a complete list of commands and their descriptions.
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: Display More Rows Options panel**

```
DEFF-R -------------------  Display More Rows Options  ------------------------
Command  ==>  

Specify the following options.
Press ENTER to continue or press END or CANCEL to exit.

Save table changes . . . . Y    Y/N save changes. N will discard changes
made since the last SAVE.
(this is ignored in table BROWSE mode)

Current range of rows displayed is: 1 .. 300
Begin next display with row . . . . 301
```

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 on page 229 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, respond to the Confirm Updates popup.

- 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 is efficient if you need to make several changes to the initial SELECT statement or 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 “Methods for invoking the data editing function” on page 223.

2. On the Edit DB2 Table Options panel, set option values for editing data. For information, see “Setting options for editing data” on page 223.

   **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.

   One of the following panels is displayed:
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: COPY From DB2 Table Options panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>COPY From DB2 Table Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the following options, then press ENTER to read table rows. Press END or CANCEL to abandon the copy.</td>
<td></td>
</tr>
<tr>
<td>Copy source table . . . . . . QZU.QZUT01_DCII9S05</td>
<td></td>
</tr>
<tr>
<td>Edit COPY select statement . . Y Display and edit copy select statement.</td>
<td></td>
</tr>
<tr>
<td>Save/Retrieve Select . . . . . N (S/R/N) N-no action</td>
<td></td>
</tr>
<tr>
<td>S-save current select statement in SQL table using the select statement name</td>
<td></td>
</tr>
<tr>
<td>R-retrieve list of saved select statements matching pattern in select statement name</td>
<td></td>
</tr>
<tr>
<td>Select statement name . . . .</td>
<td></td>
</tr>
<tr>
<td>Number of rows to Copy . . . . 0 0-99999999 rows to copy. 0 = no limit. Approximate upper limit is shown.</td>
<td></td>
</tr>
</tbody>
</table>

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 “Supported wildcards in qualifiers” on page 40.

- 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 **Figure 98 on page 231**, 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: Select Statement Specification panel**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>ORDER</th>
<th>OPER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_1</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_2</td>
<td>SMALLINT</td>
<td>2</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_3</td>
<td>CHAR</td>
<td>12</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_4</td>
<td>CHAR</td>
<td>7</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_5</td>
<td>CHAR</td>
<td>4</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_6</td>
<td>VARCHAR</td>
<td>55</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_7</td>
<td>DECIMAL</td>
<td>31,20</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_8</td>
<td>DECIMAL</td>
<td>11,2</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_9</td>
<td>SMALLINT</td>
<td>2</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_10</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_11</td>
<td>FLOAT</td>
<td>4</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_12</td>
<td>FLOAT</td>
<td>8</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_13</td>
<td>DATE</td>
<td>4</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_14</td>
<td>TIME</td>
<td>3</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_15</td>
<td>TIMESTMP</td>
<td>10</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_16</td>
<td>VARCHAR</td>
<td>30</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_17</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_18</td>
<td>CHAR</td>
<td>20</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_19</td>
<td>CHAR</td>
<td>24</td>
<td>A</td>
<td></td>
<td>=</td>
</tr>
<tr>
<td>COLUMN_20</td>
<td>INTEGER</td>
<td>4</td>
<td>A &gt; 20</td>
<td></td>
<td>=</td>
</tr>
</tbody>
</table>

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

**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 185.

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: Edit DB2 Table panel showing rows that were copied

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF</td>
<td>CSR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>*INS</th>
<th>COLUMN_1</th>
<th>COLUMN_2</th>
<th>COLUMN_3</th>
<th>COLUMN_4</th>
<th>COLUMN_5</th>
<th>COLUMN_6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1818599</td>
<td>1214064</td>
<td>4064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818629</td>
<td>1130064</td>
<td>0064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818687</td>
<td>1139064</td>
<td>9064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818706</td>
<td>1141064</td>
<td>1064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818722</td>
<td>1145064</td>
<td>5064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818733</td>
<td>1146063</td>
<td>6063</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818754</td>
<td>1148064</td>
<td>8064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818758</td>
<td>1148065</td>
<td>8064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818781</td>
<td>1160064</td>
<td>0064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818810</td>
<td>1164064</td>
<td>4064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818836</td>
<td>1171664</td>
<td>0664</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818895</td>
<td>1177066</td>
<td>7066</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818902</td>
<td>1182064</td>
<td>2064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818932</td>
<td>1184064</td>
<td>4064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1818944</td>
<td>1185064</td>
<td>4064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1819054</td>
<td>1195064</td>
<td>4064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1447973</td>
<td>1655233</td>
<td>5233</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1819036</td>
<td>1209064</td>
<td>9064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1448522</td>
<td>0773625</td>
<td>3625</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1448585</td>
<td>0774002</td>
<td>3625</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1819322</td>
<td>1185064</td>
<td>4064</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1448585</td>
<td>0774002</td>
<td>3625</td>
<td>DEFENDANT ARRAIGN</td>
<td></td>
<td></td>
<td></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

The copy table rows option is efficient if you want to copy 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.

To copy data by specifying an option on the Edit DB2 Table Options panel

1 Invoke a data editing session for the target table or view.

For more details, see “Methods for invoking the data browsing function” on page 216.

2 On the Edit DB2 Table Options panel, set option values for editing data.
Note

For more details, see “Setting options for editing data” on page 223. 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 (see Figure 101 on page 234).

**Figure 101: Copy Table Rows Specifications panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DEFF-R</code></td>
<td>Copy Table Rows Specifications</td>
</tr>
<tr>
<td>Specify a table name or pattern for the source table and target table. Select the other options and press enter to copy rows. Press END or CANCEL to abandon the copy.</td>
<td></td>
</tr>
<tr>
<td>Insert rows target table . . QZU.QZUT01_DCII5502</td>
<td></td>
</tr>
<tr>
<td>Select rows source table . . RDACRJ.DEF_QZUT01</td>
<td></td>
</tr>
<tr>
<td>Edit subselect statement . . Y</td>
<td>Edit SQL select statement to specify WHERE clause values</td>
</tr>
<tr>
<td>Edit insert statement . . Y</td>
<td>Edit SQL insert statement</td>
</tr>
<tr>
<td>Delete all rows first . . N</td>
<td>Delete all rows from target table before executing the insert statement?</td>
</tr>
</tbody>
</table>

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 “Supported wildcards in qualifiers” on page 40.

In Figure 101 on page 234, the following options have been selected:

- To copy from a table
- To edit the SELECT statement
- To edit the INSERT statement

5 When you have specified all of the options, press **Enter**.
The Select Statement Specification panel is displayed.

6 Customize the SELECT statement by modifying the selected columns and values on the panel (see Figure 102 on page 235).

*Tip*

Press HELP to display example specifications.

**Figure 102: Select Statement Specification panel**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>LENGTH</th>
<th>ORDER</th>
<th>OPER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_1</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_2</td>
<td>SMALLINT</td>
<td>2</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_3</td>
<td>CHAR</td>
<td>12</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_4</td>
<td>CHAR</td>
<td>7</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_5</td>
<td>CHAR</td>
<td>12</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_6</td>
<td>VARCHAR</td>
<td>55</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_7</td>
<td>DECIMAL</td>
<td>31,20</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_8</td>
<td>DECIMAL</td>
<td>11,20</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_9</td>
<td>SMALLINT</td>
<td>2</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_10</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_11</td>
<td>FLOAT</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_12</td>
<td>FLOAT</td>
<td>8</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_13</td>
<td>DATE</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_14</td>
<td>TIME</td>
<td>3</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_15</td>
<td>TIMESTAMP</td>
<td>10</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_16</td>
<td>VARCHAR</td>
<td>30</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_17</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_18</td>
<td>CHAR</td>
<td>20</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_19</td>
<td>CHAR</td>
<td>24</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
<tr>
<td>COLUMN_20</td>
<td>INTEGER</td>
<td>4</td>
<td>A</td>
<td>=</td>
<td>VALUE</td>
</tr>
</tbody>
</table>

*Note*

You can substitute host variables for the Value specifications in the SELECT statement. For more information, see “Using host variables in a search” on page 185.

7 Press END.

The Copy Table Rows Specifications panel is displayed. You can specify whether to change other options on the panel.

8 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 (see Figure 103 on page 236).

**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_DCII5S02 ( 
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,
```

9  Save the changes to the INSERT statement and close the ISPF edit panel.

The Copy Table Rows Specifications panel is displayed.

10 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.

### Where to go from here

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>“Creating objects” on page 239</td>
</tr>
<tr>
<td>Drop and recover objects</td>
<td>“Dropping and recovering objects” on page 261</td>
</tr>
</tbody>
</table>
Creating objects

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 “Working with lists and searches” on page 173 before reading this chapter.

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.

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 90.

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 more information, see “Space requirement estimates based on user-specified values” on page 240.
Space requirement estimates 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 do not need to run BMCSTATS before using SSE.

For example, for a non-partitioned table space estimate, you can change the emboldened fields in Figure 104 on page 240. 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.

**Figure 104: Tablespace Estimation panel**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Estimated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pagesize</td>
<td>16 (4,8,16,32)</td>
<td>Pages . . . .</td>
</tr>
<tr>
<td>Segsize</td>
<td>4 (0-64)</td>
<td>Space (Trks) .</td>
</tr>
<tr>
<td>Maxrows</td>
<td>255 (0-255)</td>
<td>Space (KB) .</td>
</tr>
<tr>
<td>Dssize</td>
<td>1</td>
<td>Extents . . .</td>
</tr>
<tr>
<td>Maxrows</td>
<td>255 (0-255)</td>
<td>Data Sets . .</td>
</tr>
<tr>
<td>Pctfree</td>
<td>0 (0-99)</td>
<td>Data Sets . .</td>
</tr>
<tr>
<td>Freepage</td>
<td>0 (0-255)</td>
<td>Data Sets . .</td>
</tr>
<tr>
<td>Compress</td>
<td>N (Y or blank)</td>
<td>Data Sets . .</td>
</tr>
<tr>
<td>Priqty</td>
<td>2000</td>
<td>Alloc Unit . .</td>
</tr>
<tr>
<td>Secqty</td>
<td>100</td>
<td>Device Type . .</td>
</tr>
<tr>
<td>Avg Row</td>
<td>1</td>
<td>Rows/Page . .</td>
</tr>
</tbody>
</table>

Estimating space requirements for a table space

Use SSE to estimate space requirements for non-partitioned, partitioned, and partition-by-growth table spaces.

You can perform the following estimations of space:
Non-partitioned table space ("To estimate space requirements for a non-partitioned table space" on page 241)

Partitioned table space ("To estimate space requirements for a partitioned table space" on page 242)

Partition-by-growth table space ("To estimate space requirements for a partition-by-growth table space" on page 242)

**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**.

3. Estimate space requirements for the entire table space or for individual tables as follows:
   
   a. Change any of the modifiable values (see “Space requirement estimates based on user-specified values” on page 240) 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 additional 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 on page 241.

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 the figure in “Space requirement estimates based on user-specified values” on page 240) as needed to correspond to your table space.

      **Note**
      For more information about a specific field, press **F1**.

   b. Press **Enter** to update the **Estimated** fields:
      
      - The **Estimated** box on the right side of the panel displays estimates for the **entire table space**.
      
      - The **Estimated** list in the bottom right corner shows estimates **per partition**.

      **Note**
      The bottom of the panel shows more information about the table. In the **Cmd** column, you can enter **I** to insert, **R** to repeat, or **D** to delete lines.

4. *(optional)* If you want to see additional estimates based on different values, repeat Step 3 on page 242.

      **Note**
      This space estimation function does not support the Average Length feature for table columns.

5. When finished, press **END** to exit.

To estimate space requirements for a partition-by-growth table space

1. From the command prompt, type **SSE** and press **Enter**.

2. On the Space Estimation panel, enter **TS** in the **Object Type** field, and **G** in the **Tablespace Type** field. Then, press **Enter**.

3. Estimate space requirements for the entire table space or for individual partitions as follows:
a. Change any of the modifiable values (highlighted in the figure in “Space requirement estimates based on user-specified values” on page 240) as needed to correspond to your table space.

Note
For more information about a specific field, press F1.

b. Press Enter to update the Estimated fields:

- The Estimated box on the right side of the panel displays estimates for the entire table space.
- The bottom of the panel shows more information about the table itself.

4. (optional) If you want to see additional estimates based on different values, repeat Step 3 on page 242.

5. When finished, press END to exit.

Estimating space requirements for an index

You can use SSE to estimate space requirements for an index.

Note
Space estimation is not available for XML objects, 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 the figure in “Space requirement estimates based on user-specified values” on page 240) as needed to correspond to your index.

Note
For more information about a specific field, press F1.

b. Press Enter to update the Estimated fields:
The Estimated box on the right side of the panel displays estimates for the index.

The Estimated list in the bottom right corner shows estimates for the index.

**Note**
The bottom of the panel shows more information about the index. In the **Cmd** column, you can enter **I** to insert, **R** to repeat, or **D** to delete lines.

4  *(optional)* If you want to see additional estimates based on different values, repeat Step 3 on page 243.

5  When finished, press END to exit.

## Using an existing object as a model to create objects

You can use the design of an existing DB2 object as a *model* to create a new object. CATALOG MANAGER makes it easy for you to change only the attributes that should differ from the model.

### To create objects

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 object qualifiers” on page 40.

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).

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 in a DB2 Version 10 subsystem to create a table from an existing table model.

The example completes these procedures:

1. “To define the table” on page 245
2. “To define additional attributes” on page 246
3. “To define the table columns” on page 247
4. “To create and edit table constraints” on page 250
5. “To edit the materialized query table options” on page 251
6. “To edit the partitions” on page 251
7. “To define the organization” on page 252
8. “To define the access control” on page 252
9. “To generate SQL” on page 252

**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 45.
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 on page 246). The displayed attribute values match those of the table that you are using as a model.

**Figure 105: Create/Alter Table panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Create/Alter Table</th>
<th>1 to 19 of 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate SQL</td>
<td>N</td>
<td>Y to generate SQL</td>
</tr>
<tr>
<td>Table owner</td>
<td>QZU</td>
<td></td>
</tr>
<tr>
<td>Table name</td>
<td>QZUT00_DSC30S28</td>
<td></td>
</tr>
<tr>
<td>Database name</td>
<td>QZUDSC30</td>
<td>% to find the database</td>
</tr>
<tr>
<td>Tablespace name</td>
<td>QZUS2830</td>
<td>% to find the tablespace</td>
</tr>
<tr>
<td>Audit</td>
<td>A-All,C-Changed,N-None,&lt;blank&gt;</td>
<td></td>
</tr>
<tr>
<td>Data capture</td>
<td>Y/N/&lt;blank&gt;</td>
<td></td>
</tr>
<tr>
<td>With Restrict on Drop</td>
<td>N</td>
<td>Y/N Y-Table cannot be dropped</td>
</tr>
<tr>
<td>Global Temporary Table</td>
<td>N</td>
<td>Y/N Create Global Temporary Table</td>
</tr>
<tr>
<td>Edit additional options</td>
<td>N</td>
<td>Y/N Edit Additional options</td>
</tr>
<tr>
<td>Edit column data</td>
<td>N</td>
<td>Y/N Edit Column information</td>
</tr>
<tr>
<td>Edit comment and label</td>
<td>N</td>
<td>Y/N Edit Comment/Label information</td>
</tr>
<tr>
<td>Edit table constraints</td>
<td>N</td>
<td>Y/N Edit Table constraints</td>
</tr>
<tr>
<td>Edit MQT Options</td>
<td>N</td>
<td>Y/N Edit MQT options</td>
</tr>
<tr>
<td>Edit partitions</td>
<td>N</td>
<td>Y/N Edit Partition options</td>
</tr>
<tr>
<td>Edit organization</td>
<td>N</td>
<td>Y/N Edit Organization options</td>
</tr>
<tr>
<td>Edit access control</td>
<td>N</td>
<td>Y/N Edit Access Control options</td>
</tr>
</tbody>
</table>

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: Create/Alter Table panel

<table>
<thead>
<tr>
<th>Command ====&gt;</th>
<th>Scroll ====&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create/Alter Table</td>
<td>PAGE</td>
</tr>
<tr>
<td>Table Creator. . . . . .</td>
<td>QZU</td>
</tr>
<tr>
<td>Table Name . . . . . .</td>
<td>QZUT00_DSC30S28</td>
</tr>
<tr>
<td>Editproc . . . . . .</td>
<td>Table Edit routine</td>
</tr>
<tr>
<td>Validproc. . . . . .</td>
<td>Validation Exit routine</td>
</tr>
<tr>
<td>CCSID. . . . . . .</td>
<td>E</td>
</tr>
<tr>
<td>Volatile . . . . . .</td>
<td>N</td>
</tr>
<tr>
<td>Append . . . . . .</td>
<td>N</td>
</tr>
<tr>
<td>Temporal Table name . . .</td>
<td></td>
</tr>
</tbody>
</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 on page 247). You can use the ISPF INSERT, DELETE and REPEAT commands to increase or reduce the number of columns.

Figure 107: Columns panel

<table>
<thead>
<tr>
<th>Command ====&gt;</th>
<th>Scroll ====&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns</td>
<td>PAGE</td>
</tr>
<tr>
<td>Table Creator. . . . . .</td>
<td>QZU</td>
</tr>
<tr>
<td>Table Name . . . . . .</td>
<td>QZUT00_DSC30S28</td>
</tr>
<tr>
<td>Enter I to Insert, R to Repeat, or D to Delete a line.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cmd Name</th>
<th>Schema</th>
<th>Data Type</th>
<th>Length</th>
<th>Scale</th>
<th>Nl</th>
<th>Df</th>
<th>Opts</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_1</td>
<td>INTEGER</td>
<td>N N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_2</td>
<td>SMALLINT</td>
<td>N Y N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_3</td>
<td>CHAR</td>
<td>1 Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_4</td>
<td>VARCHAR</td>
<td>1 N Y N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_5</td>
<td>DECIMAL</td>
<td>2 N Y N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_6</td>
<td>VARCHAR</td>
<td>1 N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_7</td>
<td>CHAR</td>
<td>3 N Y N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_8</td>
<td>INTEGER</td>
<td>Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_9</td>
<td>REAL</td>
<td>N Y N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_10</td>
<td>VARCHAR</td>
<td>1 Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_11</td>
<td>DOUBLE</td>
<td>N Y N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_12</td>
<td>DATE</td>
<td>Y N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_13</td>
<td>CHAR</td>
<td>1 N N N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 **Nulls** field.

g To indicate whether a default value is placed in a column, specify a value in the **With default** 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: Column Options panel**

<table>
<thead>
<tr>
<th>Column Options</th>
<th>1 to 17 of 17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td></td>
</tr>
<tr>
<td>Table Creator. . . . . OZU</td>
<td></td>
</tr>
<tr>
<td>Table Name . . . . QZUT00_DSC3OS2B</td>
<td></td>
</tr>
<tr>
<td>Column name . . . . COLUMN_1</td>
<td></td>
</tr>
<tr>
<td>Data type Schema . . . .</td>
<td></td>
</tr>
<tr>
<td>Data type . . . . . . INTEGER</td>
<td></td>
</tr>
<tr>
<td>Length . . . . . .</td>
<td>Length of the field</td>
</tr>
<tr>
<td>Inline Length . . . .</td>
<td>Byte length of inline LOB column</td>
</tr>
<tr>
<td>Scale. . . . . . .</td>
<td># of digits to the right of the decimal pt</td>
</tr>
<tr>
<td>Units. . . . . . .</td>
<td>K-Kilobytes, M-Megabytes, G-Gigabytes</td>
</tr>
<tr>
<td>Nulls. . . . . . . N</td>
<td>Y/N N=NOT NULL</td>
</tr>
<tr>
<td>With default . . . . N</td>
<td>? for selection list</td>
</tr>
<tr>
<td>Default Value . . . .</td>
<td></td>
</tr>
<tr>
<td>Generated. . . . .</td>
<td>Always, By default</td>
</tr>
<tr>
<td>Generated Type. . . .</td>
<td>I/F-As Identity or For Each Row On Update</td>
</tr>
<tr>
<td>Subtype. . . . .</td>
<td>S-SBCS, B-Bit, M-Mixed</td>
</tr>
<tr>
<td>Edit Addtnl column opts. N</td>
<td>Y/N Edit Additional column options</td>
</tr>
</tbody>
</table>

5 Edit the column options as needed:

a Type **Y** in the **Edit Addtnl column opts** field, and press **Enter**
The Additional column options panel is displayed.

**Figure 109: Additional column options panel**

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Additional column options</th>
<th>Scroll ===&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column name.</td>
<td>COLUMN_1</td>
<td></td>
</tr>
<tr>
<td>Fieldproc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hidden</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Period</td>
<td>B/C/S/T Business or system time period</td>
<td></td>
</tr>
<tr>
<td>Edit comment and label</td>
<td>N</td>
<td>Y/N Edit Comment/Label information</td>
</tr>
<tr>
<td>Edit Column Identity</td>
<td>N</td>
<td>Y/N Edit Identity options</td>
</tr>
</tbody>
</table>

b. Accept or modify the default attribute values shown on the rest of the panel.

c. 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.

d. 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: Column Identity Information panel**

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Column Identity Information</th>
<th>Scroll ===&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column name.</td>
<td>COLUMN_1</td>
<td></td>
</tr>
<tr>
<td>Generated.</td>
<td>Always. By default</td>
<td></td>
</tr>
<tr>
<td>Start/Restart with</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Increment by</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>MinValue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxValue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cache</td>
<td>Y/N Preallocate and keep in memory</td>
<td></td>
</tr>
<tr>
<td>Cache Amount</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Cycle</td>
<td>Y/N Continue after reaching min/max</td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td>Y/N Generate in order of request</td>
<td></td>
</tr>
</tbody>
</table>
e 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: Table Constraints panel*

<table>
<thead>
<tr>
<th>Command</th>
<th>Table Constraints 1 to 6 of 6</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table Creator</td>
<td>QZU</td>
<td>PAGE</td>
</tr>
<tr>
<td>Table Name</td>
<td>QZUT00_DSC30S28</td>
<td></td>
</tr>
<tr>
<td>Edit Unique/Primary</td>
<td>N</td>
<td>Y/N Edit Unique/Primary Constraints</td>
</tr>
<tr>
<td>Edit Foreign Key</td>
<td>N</td>
<td>Y/N Edit Foreign Key Constraints</td>
</tr>
<tr>
<td>Edit Check</td>
<td>N</td>
<td>Y/N Edit Check Constraints</td>
</tr>
</tbody>
</table>

3 In the *Edit Unique/Primary* field, type Y to edit the unique and primary constraints. Press Enter.

The Table Unique and Primary Constraints panel is displayed. If the model table contains unique or primary constraints, they are listed on this panel.

a In the *Cmd* column, type I (insert), R (repeat) or D (delete).

---

**Tip**
Press HELP for a description of the fields on the panel.

b Press END to display the Table Constraints panel.

4 In the *Edit Foreign Key* field, type Y to edit foreign keys. Press Enter.

The Table Foreign Key Constraints panel is displayed. If the model table contains foreign key constraints, they are listed on this panel.

---

**Note**
To create or drop a foreign key, you must have the ALTER privilege on the parent and dependent tables.

a In the *Cmd* column, type I (insert), R (repeat) or D (delete). Press HELP for a description of the fields on the panel.

b Press END to display the Table Constraints panel.

5 In the *Edit Check* field, type Y to edit the check constraints. Press Enter.
The Table Check Constraints panel is displayed. If the model table contains check constraints, they are listed on this panel.

a In the **Cmd** column, type **I** (insert), **R** (repeat) or **D** (delete). Press HELP for a description of the fields on the panel.

b Press END to display the Table Constraints panel.

6 Press END to display the Create/Alter Table panel.

**To edit the materialized query table options**

1 In the **Edit MQT Options** field, type **Y** to edit materialized query table options, and press **Enter**.

The Materialized Query Options panel is displayed.

![Figure 112: Materialized Query Options panel](image)

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: Select Generate Text panel](image)

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**.

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 on page 253). This panel shows the statements generated by CATALOG MANAGER based on your specifications.

Figure 114: Confirm SQL panel for creating a table

<table>
<thead>
<tr>
<th>Command</th>
<th>Confirm SQL</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBDC-R</td>
<td>Confirm SQL</td>
<td>PAGE</td>
</tr>
<tr>
<td>Current SQLID</td>
<td>RDACRJ</td>
<td></td>
</tr>
<tr>
<td>Edit Options</td>
<td>Y/N Modify SQL processing options</td>
<td></td>
</tr>
<tr>
<td>Edit SQL</td>
<td>Y/N Edit SQL before executing</td>
<td></td>
</tr>
<tr>
<td>Save in SQL table</td>
<td>A/Y/R/N A/Y-Append, R-Replace</td>
<td></td>
</tr>
<tr>
<td>Name of saved SQL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save in PDS</td>
<td>Y/N Save SQL in PDS</td>
<td></td>
</tr>
<tr>
<td>PDS(member)</td>
<td>ACT.V10.DATABASE(TEST)</td>
<td></td>
</tr>
<tr>
<td>Execute SQL</td>
<td>Y/N Execute the SQL</td>
<td></td>
</tr>
</tbody>
</table>

CREATE TABLE QZU.QZUT00_DSC30S28
  (COLUMN_1 INTEGER NOT NULL WITH DEFAULT, COLUMN_2 SMALLINT NOT NULL WITH DEFAULT, COLUMN_3 CHAR(12) NOT NULL WITH DEFAULT FOR SBCS DATA, COLUMN_4 CHAR(7) NOT NULL WITH DEFAULT FOR SBCS DATA, COLUMN_5 CHAR(4) NOT NULL WITH DEFAULT FOR SBCS DATA)

1 On the Confirm SQL panel, you can edit and save the SQL to create the object and then execute it:

a (optional) From the Command line, issue the SET sqlid command to change the value of the Current SQLID field.

Note The ID shown in the Current SQLID field must have the proper authority to perform the specified SQL CREATE statement. If you hold a primary- or secondary-authorization ID that has the proper authority, you can change the Current SQLID to that authorization ID and complete the CREATE. To change the Current SQLID, use the SET command.

b (optional) In the Edit Options field, type Y to modify the default values for the options on the Confirm SQL panel. Then, press Enter.

The SQL and Confirm Options panel is displayed. Press END to return to the Confirm SQL panel.

c (optional) In the Edit SQL field, type Y or N to invoke an ISPF edit session to edit the SQL statement. Then, press Enter.
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.

d  Press END to save the SQL and return to the Confirm SQL 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) 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.

The DDL commands generate SQL to create the source objects themselves. In contrast, the CREATE command (see “Using an existing object as a model to create objects” on page 244), generates SQL to create objects that are like source objects.

To generate DDL to create objects

1. Generate an object list.
2. Specify the appropriate DDL command.
3. Generate SQL.

Table 45 on page 256 describes the different DDL commands.
Table 45: Data definition language 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 From the <strong>Command</strong> line: DDL <code>objectCode</code> <code>ownerName.objectName</code></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Generates DDL to create the objects for which the command was entered</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>For multiple objects, generates individual DDL streams</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The settings for the fields on the following options panels affect the DDL:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>— General Options: <strong>Decimal point</strong> and <strong>SQL string delimiter</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></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The BATCH keyword is valid for the DDL command. If you use the keyword in the command, you must issue the BATCH command to generate JCL. For more information, see “Generating JCL for a job in batch” on page 201</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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
<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:</td>
<td>DB</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>MQT</td>
</tr>
<tr>
<td></td>
<td>Enables you to include GRANT authorizations in the SQL</td>
<td></td>
<td>NP</td>
</tr>
<tr>
<td></td>
<td></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>PR</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 201.</td>
<td>TB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VW</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Considerations</td>
<td>Source objects</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>HDDL (continued)</td>
<td></td>
<td>The settings for the fields on the following options panels affect the HDDL:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Object Use Options: Include in HDDL and Include in HDDL commit counts</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>— SQL and Confirm Options: all fields</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>— SQL Select: all fields</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>— Switches: Define No, Cmp &gt; 32k, HDDL Auths, and Build SQLID before GRANT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you specify N for the HDDL Auths switch and later decide to generate GRANT statements without CREATEs, use the HGRANT command. In this way, you can replicate objects on another DB2 subsystem and then grant identical or edited authorizations on the replicated objects. For more information about granting authorizations, see “Managing authorizations” on page 303.</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Considerations</td>
<td>Source objects</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>
| From an object list: MDDL From the **Command** line: MDDL **objectCode ownerName.objectName** | ■ Generates DDL to create the objects for which the command was entered  
■ Applies to only a single object type  
■ For multiple objects of the same object type, generates one DDL stream for all of the objects | ■ MDDL is a wait-for-enter command. For more information, see “Issuing Wait-for-Enter commands against multiple objects” on page 55.  
■ The settings for the fields on the following options panels affect the MDDL:  
  — SQL and Confirm Options: all fields  
  — SQL Select: all fields  
■ The BATCH keyword is not valid for the MDDL command. | AL  
CX  
DB  
DT  
FN  
IM  
IX  
MK  
MQT  
NP  
PM  
SE  
SG  
SY  
TB  
TR  
TS  
VW  
XT |

**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.

“Dropping and recovering objects” on page 261 explains how the drop and recovery options in CATALOG MANAGER help you work productively and with a minimum risk of error.
Where to go from here
Dropping and recovering objects

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.

**WARNING**

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.

Use 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.

To simulate a drop

1. Set the Drop switch to N. For information, see “Setting CATALOG MANAGER switches” on page 90.

   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 264.
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 45.

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 55.)

4. Press **Enter**.

   The Confirm DROP panel is displayed (Figure 115 on page 264), providing several actions and options that relate only to the drop and drop recovery functions.

   **Figure 115: Confirm DROP panel for DROP TABLE SPACE procedure**

   ![Confirm DROP panel](image)

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.**
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.

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 on page 265). 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:</th>
</tr>
</thead>
<tbody>
<tr>
<td>QZUDAC.QZUS01AC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Object Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>.TB</td>
<td>QZU.QZUT01_DAC501</td>
</tr>
<tr>
<td>.IX</td>
<td>QZU.QZUX01_DAC501T01</td>
</tr>
<tr>
<td>.IX</td>
<td>QZU.QZUX02_DAC501T01</td>
</tr>
<tr>
<td>.IX</td>
<td>QZU.QZUX03_DAC501T01</td>
</tr>
<tr>
<td>.IX</td>
<td>QZU.QZUX04_DAC501T01</td>
</tr>
<tr>
<td>.IX</td>
<td>QZU.QZUX05_DAC501T01</td>
</tr>
<tr>
<td>.IX</td>
<td>QZU.QZUX06_DAC501T01</td>
</tr>
<tr>
<td>.IX</td>
<td>QZU.QZUX07_DAC501T01</td>
</tr>
</tbody>
</table>

*************** Bottom of data ***************

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.
For native SQL procedures, the following statement is generated:

```sql
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:

```sql
DROP PROCEDURE procedureName RESTRICT;
```

e  Press END to save the SQL and return to the Confirm DROP panel.

f  *(optional)* In the Save in SQL Table field, enter A, Y, R, or N to specify whether to save the SQL in the CATALOG MANAGER SQL_Table.

Press END to save the SQL and return to the Confirm DROP panel.

<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 SQL.

h  *(optional)* In the Save in PDS field, enter 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.

i  *(optional)* In the PDS(member) field, type the name of the PDS and member.

j  *(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.

k  *(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.
1. (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.

m. (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.

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.

n. (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 on page 268). 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 263), 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

| 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 MOTT MX NP OS PA PDD PG PL LIKE QZU% |
| DEFF-R ---------------------- TABLESPACE LIST --------------- DROP RC= 000 |

Recovering an object and its data

The following procedure describes how to recover a table space structure that was dropped by CATALOG MANAGER, ALTER, or CHANGE MANAGER and how to recover the data in the table space:

**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 264.

**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 on page 269). This list is a subset of the Drop Recovery Log.

**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.PLBALT02</td>
</tr>
<tr>
<td>2011-01-19</td>
<td>14.09</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>PLBDBA1.PLBALTER</td>
</tr>
<tr>
<td>2011-01-17</td>
<td>14.16</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>PXBQXPB.PBASPB1</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>2011-01-10</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>RDAMSL</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5.MXSS01G5</td>
</tr>
<tr>
<td>2011-01-08</td>
<td>11.25</td>
<td>RDAMSL</td>
<td>AEX010</td>
<td>TS</td>
<td>MXCDBG5.MXSS01G5</td>
</tr>
<tr>
<td>2011-01-07</td>
<td>17.44</td>
<td>RDAMSL</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 255).

   c. Go to Step 1 on page 268.

4. When the Primary Menu panel or list panel is displayed, type **S S** in the **Cmd** column beside the table space to be recovered, and then press **Enter**.
The Recovery Statements panel is displayed (Figure 119 on page 270), which shows which objects will be recovered with the table space.

**Figure 119: Recovery Statements panel**

<table>
<thead>
<tr>
<th>DEFF.R</th>
<th>---------------------</th>
<th>Recovery Statements</th>
<th>-----------</th>
<th>Row 1 to 4 of 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>==&gt;&gt;</td>
<td>Scroll</td>
<td>==&gt;&gt;</td>
<td>CSR</td>
</tr>
<tr>
<td>Enter D or S to describe or ED to edit a single recovery log row.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute recovery statements now</td>
<td>N (Y/N)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TABLESPACE=DBXNAUT.SBXNCOL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enter X on statement line to exclude it from recovery.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cmd Seq</td>
<td>Type</td>
<td>Name</td>
<td>Text</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TS</td>
<td>DBXNAUT.SBXNCOL</td>
<td>CREATE TABLESPACE SBXNCOL IN DB</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TB</td>
<td>RDABXN.TBXNCOL</td>
<td>CREATE TABLE RDABXN.TBXNCOL ( C</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TB</td>
<td>RDABXN.TBXNCOL</td>
<td>ALTER TABLE RDABXN.TBXNCOL ACTI</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TB</td>
<td>RDABXN.MBXN_MASK</td>
<td>CREATE MASK RDABXN.MBXN_MASK ON</td>
<td></td>
</tr>
</tbody>
</table>

---

**Table 46: 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, enter X in the **Cmd** column beside each object.

**Tip**

You can mark all indexes for exclusion by entering 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 on page 271 shows a Describe Audit Log Entry panel. Press END to display the Recovery Statements Panel.

**Figure 120: Describe Audit Log Entry panel for the Drop Recovery Log**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF-R</td>
<td>CSR</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FROM ACT010.RECOVERY_LOG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logts. . . . . : 2011-02-01-14.43.53.716869</td>
</tr>
<tr>
<td>Authid . . . . : RDAPXB2</td>
</tr>
<tr>
<td>Session_seq . . . : 1</td>
</tr>
<tr>
<td>Sequence . . . : 1</td>
</tr>
<tr>
<td>Drop_seq . . . : 1</td>
</tr>
<tr>
<td>DBID . . . . . : 350</td>
</tr>
<tr>
<td>OBID . . . . . : 1</td>
</tr>
<tr>
<td>PSID . . . . . : 2</td>
</tr>
<tr>
<td>Object_type . . . : TABLESPACE</td>
</tr>
<tr>
<td>Object_qual . . . : DBXNAUT</td>
</tr>
<tr>
<td>Object_name . . . : SBXNCOL</td>
</tr>
<tr>
<td>Dep_obj_typ . . . : TABLESPACE</td>
</tr>
<tr>
<td>Dep_obj_qal . . . : DBXNAUT</td>
</tr>
<tr>
<td>Dep_obj_nam . . . : SBXNCOL</td>
</tr>
<tr>
<td>Product . . . . : ACT010</td>
</tr>
<tr>
<td>Action . . . . : CREATE TABLESPACE SBXNCOL IN DBXNAUT USING STOGROUP</td>
</tr>
</tbody>
</table>

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**.

3. In the DB2 Commands panel, press **Enter** to execute the command.

   CATALOG MANAGER sends the command to DB2.

4. Press **END**.

5. In the Drop Recovery List, type **S** in the **Cmd** column beside the table space to be recovered, and then press **Enter**.

   The Recovery Statements panel is displayed.
6 Type `RECOVER` in the `Cmd` column of the `REC DATA` text line. Press `Enter`.

**Note**

If several incremental image copies exist, or if the table space had several partitions, several `REC DATA` text lines might be displayed in the Recovery Statements panel. To recover data, you need to type `RECOVER` for only one `REC DATA` line.

CATALOG MANAGER recovers only to the last image copy. If you also want to recover changes from the log, you must know the `to logrba`. `REC LRBA` provides the `to logrba`. Recovery from the log is only possible if the OBIDs did not change between dropping and recreating the objects.

CATALOG MANAGER invokes the DSN1COPY utility to recover the data that was stored in the dependent tables. The DSN1COPY Droprecovery Utility panel is displayed. The Utility Options section of the panel shows each option that you can set when using DSN1COPY. Press HELP for a description of these options.

7 In the `JCL Dataset` field, verify that the specified JCL data set name is allocated and that this is the correct data set.

8 In the `Set JCL Options` field, type `Y` to set your options for generating JCL.

   The JCL Generation Options panel is displayed. Press END to return to the DSN1COPY Droprecovery Utility panel.

9 In the `Build Job` field, type `Y` to build the JCL. Press `Enter`.

10 In the `Edit Dataset` field, type `Y` to edit the data set. Press `Enter`.

   The JCL data set is displayed in the ISPF edit panel. After editing, press END to display the DSN1COPY Droprecovery Utility panel.

11 In the `Submit` field, type `Y` to submit the JCL. Press `Enter`.

   The batch job must be completed before you can proceed.

12 When the batch job is completed, press END to display the Recovery Statements panel.

13 To restart the table space, type `START` in the `Cmd` column of the `REC DATA` text line. Press `Enter`.

14 In the DB2 Commands panel, press `Enter` to execute the command.

   CATALOG MANAGER sends the command to DB2.

15 Press END.
16 In the Drop Recovery List, type `S` in the `Cmd` column beside the table space to be recovered, and then press `Enter`.

The is displayed.

17 In the Recovery Statements panel complete the following steps to recover indexes:

*Tip*
To recover more than one index in the same table space, create a table space (TS) list, and enter `RECOVER IX ALL` in the `Cmd` column beside the table space name.

a. Remove the `X` notations from any indexes that you marked for exclusion.

b. In the `Execute recovery statements now` field, type `Y`.

c. Press `Enter`.

The table space and data recovery are complete.

**Where to go from here**

With CATALOG MANAGER you can easily generate and submit JCL for BMC and IBM utility jobs.

“Generating utility jobs” on page 275 shows you how to do perform these tasks.
Where to go from here
Generating utility jobs

This section 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.

For more information, view the Quick Course "Generating Utilities."

Available utilities

This section describes how to generate BMC and IBM DB2 utility jobs by using CATALOG MANAGER, you can initiate processing for BMC products and IBM utilities.

BMC products

In This section describes how to generate BMC and IBM DB2 utility jobs by using CATALOG MANAGER, you can initiate processing for BMC products by using the commands listed in the following table.

For information about issuing commands in CATALOG MANAGER, see “Listing and executing commands” on page 52.
Table 47: 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>BMCCHECKS</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>BMCCOPY</td>
<td></td>
</tr>
<tr>
<td>BMLOAD</td>
<td>LOADPLUS for DB2</td>
</tr>
<tr>
<td>BMCREBUILD</td>
<td>RECOVER PLUS for DB2</td>
</tr>
<tr>
<td>BMCREBUILD INDEX</td>
<td></td>
</tr>
<tr>
<td>BMCREBUILD IX</td>
<td></td>
</tr>
<tr>
<td>BMCRECOVER</td>
<td>REORG PLUS for DB2</td>
</tr>
<tr>
<td>BMCRECOVER INDEX</td>
<td></td>
</tr>
<tr>
<td>BMCRECOVER IX</td>
<td></td>
</tr>
<tr>
<td>BMCRECOVERY</td>
<td></td>
</tr>
<tr>
<td>BMCREORG</td>
<td></td>
</tr>
<tr>
<td>BMCREORG INDEX</td>
<td></td>
</tr>
<tr>
<td>BMCREORG IX</td>
<td></td>
</tr>
<tr>
<td>BMCREORG</td>
<td></td>
</tr>
<tr>
<td>BMCRESTATS</td>
<td>BMCSTATS (component of the BMC DASD MANAGER PLUS for DB2 product)</td>
</tr>
<tr>
<td>BMCRESTATS INDEX</td>
<td></td>
</tr>
<tr>
<td>BMCRESTATS IX</td>
<td></td>
</tr>
<tr>
<td>BMCRESTATS</td>
<td></td>
</tr>
<tr>
<td>BMUNLOAD</td>
<td>UNLOADPLUS for DB2</td>
</tr>
<tr>
<td>BMUNLOAD INDEX</td>
<td></td>
</tr>
<tr>
<td>BMUNLOAD IX</td>
<td></td>
</tr>
</tbody>
</table>

CATALOG MANAGER Help panels and “JCL Generation product options” on page 415 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 the following table.

For information about issuing commands in CATALOG MANAGER, see “Listing and executing commands” on page 52.
### Table 48: Commands to invoke IBM DB2 utilities

<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>REORG INDEX</td>
</tr>
<tr>
<td>REORG IX</td>
<td></td>
</tr>
<tr>
<td>REPORT</td>
<td>REPORT</td>
</tr>
<tr>
<td>REPORT INDEX</td>
<td></td>
</tr>
<tr>
<td>REPORT IX</td>
<td></td>
</tr>
</tbody>
</table>
CATALOG MANAGER Help panels and “Commands” on page 501 provide more information about using these commands. For information about the IBM DB2 utilities, see the IBM documentation.

### Multiple utility command limitations

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 maximum 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 55.

### Utility profiles

A utility profile is a file that contains customized specifications for the syntax of a utility.

Using utility profiles saves you time and helps avoid user errors because you avoid having to specify the syntax each time that you generate a job. Using utility profiles is not required; however, it is strongly recommended by BMC.

CATALOG MANAGER creates the following types of utility profiles:

- Site profiles, which apply to all users

<table>
<thead>
<tr>
<th>CATALOG MANAGER command</th>
<th>IBM DB2 utility</th>
</tr>
</thead>
<tbody>
<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>
■ User profiles, which are available to the users who create them
Minimal setup is required to enable you to use utility profiles:

■ Allocate a data set in which to save the utility profiles by using the following criteria:
  — Record format = VB (variable blocked)
  — Record length = 255

■ Specify the data set in CATALOG MANAGER options, as follows:
  — Specify the site profile data set in the TDSN installation option of the CATALOG MANAGER installation options. The default value of the TDSN option is blank. The data set can be either sequential or partitioned. If the data set is partitioned, ensure that you include the member name in the data set. For more information, see “CATALOG MANAGER installation options” on page 401.
  — Specify the user profile data set on the Datasets panel. For more information, see “Setting data set options” on page 84.

BMC recommends that you save all utility profiles in members of the same data set.

For more information, view the Quick Course "Setting Up User Utility Profiles."

Creating a user utility profile data set

This section describes the various procedures that you can use 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.
3 In the Edit Dataset names field, type **Y**.

The Datasets panel is displayed.

4 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.

---

5 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 45.

2 On the list panel, enter **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, enter **PROFILE SAVEAS DEFAULT**.

The Allocate Data Set panel is displayed.

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

9 Press **Enter**.

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.
Tip
To populate the Utility List panel with the last used profile, set the Last used prof switch. For more information, see “Working with the last used utility profile” on page 290.

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 45.

2 For a list of the commands that invoke specific utilities, see “BMC products” on page 275 and “IBM DB2 utilities” on page 276. 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 on page 282 shows the Utility List panel for the REORG PLUS utility.

**Figure 121: Utility List panel**

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

<table>
<thead>
<tr>
<th>Cmd Status</th>
<th>St#</th>
<th>Utility</th>
<th>Typ Name</th>
<th>Part Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD34.QZUS0134</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD34.QZUS0134</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD35.QZUS0135</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD35.QZUS0235</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD35.QZUS0335</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Reorplus</td>
<td>TS</td>
<td>QZUD35.QZUS0435</td>
<td></td>
</tr>
</tbody>
</table>

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 84.

   To specify a different data set name for only the job that you are building, type the data set name in the **JCL Dataset** field. The change does not persist for subsequent jobs.

2. In the **User Profile Dsn** field, verify the data set and member for user utility profiles.

3. In the **Set JCL options** field, type **Y** to view and modify the JCL options panels, and then press **Enter**.

4. After specifying the JCL options, press **END** to display the Utility List panel.

5. In the **Utility ID** field, type a unique identification for this utility operation.

   You can include from 1 to 16 characters. If you do not enter a value for this field, CATALOG MANAGER substitutes the SSID for the Utility ID in the generated JCL.
The **Utility ID** field can contain variable names which will be changed to the variable value in the generated JCL. Table 49 on page 283 shows valid variables for the Utility ID.

### Table 49: Utility ID variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;UTIL</td>
<td>Utility name, such as REORG PLUS or LOADPLUS</td>
</tr>
<tr>
<td>&amp;ACTUDB&lt;sup&gt;5,a&lt;/sup&gt;</td>
<td>Database of the first table space that is selected</td>
</tr>
<tr>
<td>&amp;ACTUTS&lt;sup&gt;5,a&lt;/sup&gt;</td>
<td>First table space that is selected</td>
</tr>
<tr>
<td>&amp;ZUSER</td>
<td>ISPF variable for User ID</td>
</tr>
<tr>
<td>&amp;ZDATE</td>
<td>ISPF variable for Date</td>
</tr>
</tbody>
</table>

<sup>a</sup> &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 512.

   **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: REORG PLUS utility panel

<table>
<thead>
<tr>
<th>Utility Panel</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF</td>
<td>Reorg Plus: QZUD34.QZUS0134 1 to 20 of 20</td>
</tr>
<tr>
<td>Command</td>
<td>Scroll ===&gt; PAGE</td>
</tr>
<tr>
<td>Utility Id</td>
<td></td>
</tr>
<tr>
<td>Part list</td>
<td></td>
</tr>
<tr>
<td>Max new parts</td>
<td>0-4096 new parts to add</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>0/1 0-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 on page 285). 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](image)

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 50 on page 285.

**Table 50: Status column values for the Utility List panel**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blank</td>
<td>You did not enter any commands for this statement.</td>
</tr>
<tr>
<td>Edited</td>
<td>You entered the ED command for this statement.</td>
</tr>
<tr>
<td>Init</td>
<td>When the Status value of the statement was blank, the product generated JCL for this statement.</td>
</tr>
<tr>
<td>Profil</td>
<td>The product initialized the options from a utility profile.</td>
</tr>
<tr>
<td>Propgt</td>
<td>The product propagated the options from the previous object.</td>
</tr>
<tr>
<td>Single</td>
<td>The product combined two or more objects into one single statement.</td>
</tr>
</tbody>
</table>
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, enter 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, enter 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 45.

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 on page 287 shows the Utility Selections Panel for table spaces.

**Figure 124: Utility Selections panel**

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Scroll ===&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility Selections for Table space             1 to 19 of 19</td>
<td></td>
</tr>
</tbody>
</table>

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.

BMC Utilities

- BMCSTATS .......... 
- CHECK PLUS (Data) .. 
- CHECK PLUS (IX) ... 
- CHECK PLUS (TS) ... 

IBM Utilities

- CHECK DATA .......... 
- CHECK INDEX ........ 
- COPY ............... 
- DSN1COMP .......... 
- DSN1COPY .......... 
- EXEC SQL .......... 

- LOAD ............... 
- LOADPLUS ........... 
- REBUILD PLUS (IX) .. 
- REBUILD INDEX ...... 
- RECOVERY PLUS (IX) 
- UNLOAD PLUS ....... 
- UNLOAD ............ 
- RECOVER PLUS (TS) . 
- REORG PLUS ....... 
- REORG ............ 
- REPORT ............ 
- RUNSTATS .......... 
- RUNSTATS (IX) ..... 

4 Select utilities in a desired order by entering 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, enter **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 281.

**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 45.
2 Issue a command to invoke a utility. For a list of the commands that invoke specific utilities, see “BMC products” on page 275 and “IBM DB2 utilities” on page 276.

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: Profile selection panel**

![Profile Selection Panel](image)

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 45.

2. Issue a command to invoke a utility. For a list of the commands that invoke specific utilities, see “BMC products” on page 275 and “IBM DB2 utilities” on page 276.

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.

To set the switch to display the last used utility profile

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.

1. Set the switch for the Last used prof field.

   For information, see “Setting CATALOG MANAGER switches” on page 90.

2. Press END to exit the panel.

To replace the last used profile with a different profile

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.

To remove the last used profile from the utility list panel

1. Enter RP in the Cmd column.

2. 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.

**To select an alternative profile from a utility profile list**

1. On the **Command** line of a utility panel, enter **PROFILE**.

2. To select a profile from the list of profiles for the selected utility type, enter **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.

**To select an alternative profile by profile ID**

1. On the **Command** line of a utility panel, enter **PROFILE SET profileID**

2. 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.

**Related Information**

- "Setting the LISTDEF and TEMPLATE data set options" on page 126
Available utilities for TEMPLATE and LISTDEF statements

The following table lists the IBM utilities for which TEMPLATE and LISTDEF statements are valid.

Table 51: 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

1. Begin the utility job by issuing a DB2 utility command against an object.

For a list of DB2 utility commands, see “IBM DB2 utilities” on page 276.
On the Utility List panel, type **ED** in the **Cmd** column of the utility statement.

Press **Enter** to display the utility options panel.

Define values for the options as needed.

In the **Listdef/Template Options** field, type **Y**.

Press **Enter**.

The Listdef/Templates panel is displayed.

**Figure 126: Listdef/Template options panel**

<table>
<thead>
<tr>
<th>Listdef/Templates: ACTADMN1.N1</th>
<th>Scroll ==&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Listdef.</strong></td>
<td><strong>Enter ? for list</strong></td>
</tr>
<tr>
<td>Other Templates - Only DDs previously selected will be generated</td>
<td><strong>Enter ? for selection list</strong></td>
</tr>
<tr>
<td>PunchDDN template.</td>
<td><strong>Enter ? for selection list</strong></td>
</tr>
<tr>
<td>UNLDDN template.</td>
<td><strong>Enter ? for selection list</strong></td>
</tr>
</tbody>
</table>

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:
  1. In the **Listdef** field, type **?** (question mark).
  2. Press **Enter** to display the selection list.

**Figure 127: Listdef selection list panel**

<table>
<thead>
<tr>
<th>Listdef/Templates: ACTADMN1.N1</th>
<th>Scroll ==&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Listdef.</strong></td>
<td><strong>?</strong> (Enter ? for list)</td>
</tr>
<tr>
<td><strong>Listdef</strong></td>
<td><strong>Row 1 to 7 of 37</strong></td>
</tr>
<tr>
<td>_ COPY13PART2</td>
<td>COPY13PART2</td>
</tr>
<tr>
<td>_ INCP1302</td>
<td>INCP1302</td>
</tr>
<tr>
<td>_ INCP1702</td>
<td>INCP1702</td>
</tr>
<tr>
<td>_ LAEXTSALL</td>
<td>LAEXTSALL</td>
</tr>
<tr>
<td>_ LALLIXINACT</td>
<td>LALLIXINACT</td>
</tr>
<tr>
<td>_ LALLIXOX12</td>
<td>LALLIXOX12</td>
</tr>
<tr>
<td>_ LBMCACT62</td>
<td>LBMCACT62</td>
</tr>
</tbody>
</table>

- Type **S** beside the applicable LISTDEF name.

- Press **Enter** to display the Listdef/Template Options panel. The **Listdef** field is populated with the selected LISTDEF statement name.

Select templates by using one of the following procedures:
In the PunchDDN template or UNLDDN template field, enter the desired statement name.

Select from a list of statement names. Use a procedure similar to Step 7 on page 293.

Press END to display the utility panel.

Where to go from here

The sections 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>“Managing authorizations” on page 303</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>“Customizing CATALOG MANAGER command access” on page 329</td>
</tr>
<tr>
<td>Browse and purge logs that CATALOG MANAGER maintains.</td>
<td>“Maintaining logs” on page 345</td>
</tr>
</tbody>
</table>
Issuing DB2 commands

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

For a list of DB2 commands, see “Commands” on page 501.

For more information, view the Quick Course "Using DB2 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.

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 Prompts panel.

**Figure 128: DB2 Command Prompts panel**

DEEG-R --------------------- DB2 Command Prompts ---------------------------

Command ===> Notice the order has recently changed

```
1. ACCESS DATABASE * 15. DISPLAY LOG  29. SET SYSPARM
2. ALTER BUFFERPOOL 16. DISPLAY PROCEDURE 30. START DATABASE
3. ALTER GROUPBUFFERPOOL * 17. DISPLAY RLIMIT 31. START DDF
4. ALTER UTILITY 18. DISPLAY THREAD 32. START FUNCTION SPEC
5. ARCHIVE LOG 19. DISPLAY TRACE 33. START PROCEDURE
6. CANCEL THREAD 20. DISPLAY UTILITY 34. START RLIMIT
* 7. DISPLAY ARCHIVE 21. MODIFY TRACE 35. START TRACE
8. DISPLAY BUFFERPOOL 22. RECOVER BSDS 36. STOP DATABASE
9. DISPLAY DATABASE 23. RECOVER INDOUBT 37. STOP DDF
10. DISPLAY DDF 24. RECOVER POSTPONED 38. STOP FUNCTION SPEC
11. DISPLAY FUNCTION SPEC 25. RESET GENERICLU 39. STOP PROCEDURE
12. DISPLAY GROUP 26. RESET INDOUBT 40. STOP RLIMIT
13. DISPLAY GROUPBUFFERPOOL 27. SET ARCHIVE 41. STOP TRACE
14. DISPLAY LOCATION 28. SET LOG 42. 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 on page 296.

**Figure 129: Start Database panel**

AUBDYN                        Start Database                        1 to 9 of 9

Command ===> Start Database

Scroll ===> PAGE

Process type . . . . . N G-Generate only, E-Generate and Execute
Database name. . . . . . Name of the database
Tablespace name. . . . Name of the tablespace
Part list/range. . . . .
Clones . . . . . . . . . . . N Process CLONE objects
Access . . . . . . . . . . <blank> RW, RO, UT, FORCE
Additional options . .

**Note**

Regardless of the command that you select, CATALOG MANAGER provides the Process type field as the first field on the panel.

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 52 on page 297.
Table 52: 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, enter the keyword and value in the Additional options field.

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 301.

e Press Enter.

   CATALOG MANAGER displays the DB2 Commands panel Figure 130 on page 298. 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 3 on page 296, 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 3 on page 296, CATALOG MANAGER displays the result of the executed command.

Figure 130: DB2 Commands panel

Using model commands

CATALOG MANAGER displays the DB2 commands that you generate and execute on the DB2 Commands panel.

To display the model commands

1. From the Primary Menu, enter DB2COMMAND on the Command line.

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 execute a generated command

1. On the Current Command line, change the asterisk (*) to a dash (-) and press Enter.

To save the current command

1. In the Model Commands line, enter 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**

1. In the **Model Commands** line, type over the command with blanks.

**To change the format of the panel**

1. From the **Command** line on the DB2 Commands panel, enter the commands shown in Table 53 on page 299 to change the format of the panel.

   CATALOG MANAGER saves these commands in your ISPF profile.

### Table 53: Commands used on DB2 Commands panel

<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>

**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

**Note**

You can also enter the command as input to batch job.
Figure 131 on page 300 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 54 on page 300 describes the variables and optional keyword for the command.

**Table 54: DB2 command variables**

<table>
<thead>
<tr>
<th>Variable or keyword</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmd</td>
<td>Specifies the DB2 command</td>
<td>For a list of DB2 commands, see “Commands” on page 501.</td>
</tr>
<tr>
<td>type</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>identifier</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>keywords</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 295.</td>
</tr>
<tr>
<td>-profile</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 301.</td>
</tr>
<tr>
<td>BATCH</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 201.</td>
</tr>
</tbody>
</table>
DB2 command profiles

A DB2 command profile is a file that contains customized specifications for the syntax of a DB2 command.

Using a DB2 command profile saves you time and helps avoid user errors because you avoid having to specify the syntax each time that you generate the syntax and execute a command.

When you save command syntax to a DB2 command profile by using the PROFILE command, CATALOG MANAGER assigns a profile ID, and stores the ID, the DB2 command keywords, and the keyword values in a utility profile. (For more information about utility profiles, see “Utility profiles” on page 278.)

Figure 132 on page 301 shows the syntax of the PROFILE command.

Table 55: Command options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVEAS profileID</td>
<td>Saves the profile to a specific profile</td>
</tr>
<tr>
<td></td>
<td>You can use a maximum of 64 characters, including spaces, in the profile ID. The characters &lt;, &gt;, ?, and / are not permitted.</td>
</tr>
<tr>
<td>SET profileID</td>
<td>Establishes a profile</td>
</tr>
<tr>
<td></td>
<td>Using the PROFILE SET command to establish a profile will reset your current profile to the values defined in the profile.</td>
</tr>
<tr>
<td>SAVE</td>
<td>Saves the DB2 command profile</td>
</tr>
<tr>
<td></td>
<td>After you establish a DB2 command profile with the PROFILE SET command, you can issue the PROFILE SAVE command to save the profile.</td>
</tr>
</tbody>
</table>

Figure 133 on page 301 shows an example of the RDACRJSTARTDB DB2 command profile in a utility profile.

Figure 133: DB2 command profile

```
<?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>
      <clone>N</clone>
  </profile>
</profiles>
```
Where to go from here

The sections 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>“Managing authorizations” on page 303</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>“Customizing CATALOG MANAGER command access” on page 329</td>
</tr>
<tr>
<td>Browse and purge logs that CATALOG MANAGER maintains.</td>
<td>“Maintaining logs” on page 345</td>
</tr>
</tbody>
</table>
Managing authorizations

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.

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, view the Quick Course "Managing Authorizations."
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, enter **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 45.
2 In the **Cmd (C)** column beside the name of the table on which you want to grant privileges, type **GRANT (GR)** (Figure 134 on page 305). You can specify any number of tables.

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

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

<table>
<thead>
<tr>
<th><strong>C</strong></th>
<th><strong>Table Name</strong></th>
<th><strong>Database</strong></th>
<th><strong>Tbspace</strong></th>
<th><strong>ColsPK</strong></th>
<th><strong>Type</strong></th>
<th><strong>Rows</strong></th>
<th><strong>Pages</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QZU.QZUT01_DACS03</td>
<td>QZUDAC</td>
<td>QZUS03AC</td>
<td>21</td>
<td>T</td>
<td>65K</td>
<td>4376</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DACS04</td>
<td>QZUDAC</td>
<td>QZUS04AC</td>
<td>21</td>
<td>T</td>
<td>65K</td>
<td>21K</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DACS05</td>
<td>QZUDAC</td>
<td>QZUS05AC</td>
<td>21</td>
<td>T</td>
<td>65K</td>
<td>21K</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DACS06</td>
<td>QZUDAC</td>
<td>QZUS06AC</td>
<td>21</td>
<td>T</td>
<td>65K</td>
<td>21K</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DACS07</td>
<td>QZUDAC</td>
<td>QZUS07AC</td>
<td>21</td>
<td>T</td>
<td>65K</td>
<td>21K</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DACS08</td>
<td>QZUDAC</td>
<td>QZUS08AC</td>
<td>21</td>
<td>T</td>
<td>65K</td>
<td>21K</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DACS09</td>
<td>QZUDAC</td>
<td>QZUS09AC</td>
<td>21</td>
<td>T</td>
<td>975K</td>
<td>304K</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DA1S01</td>
<td>QZUDA1</td>
<td>QZUS01A1</td>
<td>21</td>
<td>T</td>
<td>2036</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DA1S02</td>
<td>QZUDA1</td>
<td>QZUS02A1</td>
<td>21</td>
<td>T</td>
<td>2036</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DA1S03</td>
<td>QZUDA1</td>
<td>QZUS03A1</td>
<td>21</td>
<td>T</td>
<td>2036</td>
<td>129</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DA1S04</td>
<td>QZUDA1</td>
<td>QZUS04A1</td>
<td>20</td>
<td>0</td>
<td>2007</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DA1S05</td>
<td>QZUDA1</td>
<td>QZUS05A1</td>
<td>2</td>
<td>0</td>
<td>46K</td>
<td>245</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DB1S01</td>
<td>QZUDB1</td>
<td>QZUS01B1</td>
<td>11</td>
<td>0</td>
<td>120K</td>
<td>6150</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DB1S02</td>
<td>QZUDB1</td>
<td>QZUS02B1</td>
<td>11</td>
<td>0</td>
<td>120K</td>
<td>750</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DB1S03</td>
<td>QZUDB1</td>
<td>QZUS03B1</td>
<td>11</td>
<td>0</td>
<td>120K</td>
<td>3033</td>
</tr>
<tr>
<td></td>
<td>QZU.QZUT01_DB1S04</td>
<td>QZUDB1</td>
<td>QZUS04B1</td>
<td>11</td>
<td>0</td>
<td>120K</td>
<td>1505</td>
</tr>
</tbody>
</table>

3 Press **Enter**.

The Grant Table Privileges panel is displayed.

**Figure 135: Grant Table Privileges panel**

<table>
<thead>
<tr>
<th><strong>Cmd</strong></th>
<th><strong>Creator.TBname</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>QZU.QZUT01_DACS03</td>
</tr>
</tbody>
</table>

**************************************** Bottom of data ***********************************************************
Note

Authorizations that are granted on multiple objects by using a wildcard character are not automatically available to objects that are created subsequently, even if those objects match the wildcard specification. To issue the same set of authorizations for a new object, use the CATALOG MANAGER COPYAUTHS command.

For more detail, see “Granting privileges by issuing the COPYAUTHS command” on page 313.

4 Specify the authorization IDs and the roles to which you want the privilege granted:

- To identify the grantees, in the AUTHIDs field, enter a maximum of 10 authorization IDs.
- To identify the roles, in the ROLEs field, enter 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, enter 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: Confirm SQL panel for granting table privileges**

<table>
<thead>
<tr>
<th>Command</th>
<th>Current SQLID</th>
<th>Edit Options</th>
<th>Edit SQL</th>
<th>Save in SQL table</th>
<th>Name of saved SQL</th>
<th>Save in PDS</th>
<th>Execute SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RDACRJ</td>
<td>Y/N</td>
<td>Y/N</td>
<td>N</td>
<td>20101021_115552</td>
<td>N</td>
<td>Y/N</td>
</tr>
</tbody>
</table>

---

**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.

---

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**

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.

d. Press END to save the SQL and return to the Confirm SQL 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>
</tbody>
</table>
To perform this action | Type
--- | ---
Replace the SQL in the SQL_Table | R
Discard the SQL | N

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 Execute SQL field, type Y to execute the SQL displayed on the Confirm SQL panel. 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 45.

3 Enter the HGRANT command by using one of the following options:
   - In the Cmd column of the list, enter HGRANT beside the object that you want to use as a model. Then, press Enter.
   - On the Command line, type the command using the following syntax and press Enter: HGRANT objectType objectName BATCH

Replace the variables as follows:

— objectType represents one of the following values DB, TS, TB, or VW.
—objectName represents the fully qualified name of an existing database, table space, table, or view (for example, ACGRNT1.N1).

—Add BATCH if you want to save the command for inclusion in a batch job.

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 on page 309). This panel contains the generated GRANT statements for the source object and its dependents.

Figure 137: Confirm SQL panel for authorization processing

<table>
<thead>
<tr>
<th>DBDC-R</th>
<th>Confirm SQL</th>
<th>1 to 13 of 13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>Scroll ===&gt; PAGE</td>
<td></td>
</tr>
<tr>
<td>Current SQLID . . . . . . .</td>
<td>RDACRJ</td>
<td></td>
</tr>
<tr>
<td>Edit Options . . . . . . .</td>
<td>N</td>
<td>Y/N Modify SQL processing options</td>
</tr>
<tr>
<td>Edit SQL . . . . . . .</td>
<td>N</td>
<td>Y/N Edit SQL before executing</td>
</tr>
<tr>
<td>Name of saved SQL . . . . .</td>
<td>20110121_115553</td>
<td></td>
</tr>
<tr>
<td>Save in SQL table . . . . .</td>
<td>N</td>
<td>A/Y/R/N A/Y-Append, R-Replace</td>
</tr>
<tr>
<td>Save in PDS . . . . . . .</td>
<td>N</td>
<td>Y/N Save SQL in PDS</td>
</tr>
<tr>
<td>PDS(member) . . . . . . .</td>
<td>ACT.V10.DATABASE(TEST)</td>
<td></td>
</tr>
<tr>
<td>Execute SQL . . . . . . .</td>
<td>N</td>
<td>Y/N Execute the SQL</td>
</tr>
</tbody>
</table>

--- SQL ---

GRANT DBADM ON DATABASE ACTGRNT1 TO ROHLQT WITH GRANT OPTION;
GRANT SELECT ON TABLE JAS3.PS_EN_TASK_RESRC TO RDATLF6;
GRANT SELECT ON TABLE JAS3.PS_EN_TASK_RESRC TO RDATLF7;
GRANT ALTER ON TABLE JAS3.PS_EN_TASK_RESRC TO CATTB02;
GRANT ALTER ON TABLE JAS3.PS_EN_TASK_RESRC TO CATTB777;
GRANT REFERENCES ON TABLE JAS3.PS_EN_TASK_RESRC TO CATTBREF;
GRANT SELECT ON TABLE JAS3.SDFSDFSDF TO RDATLF6;
GRANT SELECT ON TABLE JAS3.SDFSDFSDF TO RDATLF7;
GRANT SELECT ON TABLE RDAKJT.ACTGRNT TO RDAPXB4;
GRANT REFERENCES ON TABLE RDAKJT.ACTGRNT TO RDAPXB3;
GRANT ALTER, DELETE, INDEX, INSERT, SELECT, UPDATE, TRIGGER,
REFERENCES ON TABLE RDAKJT.ACTGRNT TO RDAPXB4;

4 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**.

d  Press **END** to save the SQL and return to the Confirm SQL 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.

To import the PDS member to another subsystem as an entry in the SQL_Table, see “Importing the SQL in another subsystem” on page 310.

i  *(optional)* In the **Execute SQL** field, type **Y** to execute the SQL displayed on the Confirm SQL panel. Then, press **Enter**.

The SQL Progress Indicator panel is displayed. The panel automatically refreshes to display the status of the SQL that is being executed.

**Importing the SQL in another subsystem**

This procedure describes how to import the HGRANT SQL that you saved to a PDS member to another subsystem as an entry in the SQL_Table.
1. Complete the steps in “Granting privileges on a hierarchy of DB2 objects” on page 308. 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 157.

4. On the target subsystem, on the Command line of the Primary Menu panel or an object list panel, enter IMPORT.

5. Press Enter.

   The Import SQL from a PDS panel is displayed.

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

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Import SQL from a PDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset imported from</td>
<td>Member name or pattern for members to be copied</td>
</tr>
<tr>
<td>Member pattern . . .</td>
<td>SQL Member owner . .</td>
</tr>
<tr>
<td>SQL Member prefix</td>
<td>SQL Member prefix for SQL Table</td>
</tr>
<tr>
<td>Overwrite like names</td>
<td>Overwrite like names</td>
</tr>
</tbody>
</table>

6. In the Dataset imported from field, enter the data set name that includes the PDS member that contains the HGRANT SQL.

7. In the Member pattern field, enter the PDS member name or a pattern that includes a wildcard.

8. *(optional)* In the SQL Member owner field, enter 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, enter 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, enter Y to replace a like-named SQL_Table entry on the current subsystem when the PDS is imported.

11. Press Enter.

   If a match for the PDS member name or pattern is found, a selection list of member names is displayed.
12 Enter 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, enter SQL on the Command line.

2 Press Enter.

The SQL Table List panel is displayed.

3 In the Cmd column, enter 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 157.

6 On the Command line, enter SQL.

7 Press Enter.

The SQL Table List panel on the target subsystem is displayed.

8 On the Command line, enter PASTE membername.

**Tip**

To view the source SQL and membername, enter CLIPBOARD on the Command line. Then press Enter. Press END to display the SQL_Table panel.
9 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 45.

2 In the User Authorizations List panel, enter 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, enter the equal sign (=) in the Cmd column beside them.

4 Press Enter.
The Copy User Authorizations panel is displayed.

**Figure 139: Copy User Authorizations panel**

<table>
<thead>
<tr>
<th>Command ===&gt;</th>
<th>Copy User Authorizations</th>
<th>Scroll ===&gt; PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate copyauth grant SQL... N       Y to generate SQL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Include implicit privileges... N       Y to include implicit privileges</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Privilege type to include... UA/RA/DA/TA/PA/GA/FA/SA/blank=all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy From... QZU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Swap with</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy From Copy To</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The **Copy To** field identifies the user IDs that were selected as the targets for the authorizations in *Step 3 on page 313*. If you did not include target user IDs, then the **Copy To** field is blank.

5  In the **Copy To** field, type the target user IDs. You can specify up to 10 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.

   c  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, enter Y to grant implicit privileges from the source user ID to the target user IDs.
8 In the **Privilege type to include** field, enter the code of the object type for which you want to copy authorizations.

9 In the **Generate copyauth grant SQL** field, enter **Y** to generate SQL.

10 Press **Enter**.

The Confirm SQL panel is displayed.

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Confirm SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current SQLID</td>
<td>RDACRJ</td>
</tr>
<tr>
<td>Edit Options</td>
<td>N</td>
</tr>
<tr>
<td>Edit SQL</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></td>
</tr>
<tr>
<td>Execute SQL</td>
<td>N</td>
</tr>
<tr>
<td>GRANT DBADM ON DATABASE QZUDSC30 TO RDACRJ ;</td>
<td></td>
</tr>
</tbody>
</table>

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, enter **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, enter **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, enter 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 310.

h  *(optional)* In the **Execute SQL** field, enter 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 45.

2  Follow the steps in “Copying user ID privileges” on page 313.
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.

The BATCH parameter can be used with the DCL command. “Generating JCL for a job in batch” on page 201 explains the procedure for generating a job by using the DESCRIBE command. You can use a similar procedure by substituting the DCL command.

**Note**
The generated SQL includes only privileges that are explicitly granted.

**To grant privileges**

1. Issue the DCL line command against one of the source object in Table 56 on page 317.

**Table 56: 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>
</tbody>
</table>
Revoking privileges

CATALOG MANAGER cannot be used to circumvent the DB2 rules for controlling access to data.

Rules for revoking authorizations can be summarized as follows:

- Authorizations can be revoked only by a user ID with SYSADM or SYSCTRL authority, or by the user ID that granted the authorization.
- Authorizations that are granted implicitly cannot be revoked. For example, you cannot revoke the authorization to create a table from a user ID with DBADM authority because creating tables is a privilege that is implicitly available to DBADM authority.
- User IDs cannot revoke authorizations from themselves.
- No user ID (including those with SYSADM or SYSCTRL authority) can revoke privileges on a DB2 resource from the creator of the resource.
- Replicate authorizations that were granted to the same user ID from more than one grantor remain effective until revoked by each grantor.

Preservation of 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 of the REVOKE command 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 201.

### To generate the cascade report

1. Create a system user list or a user list. For information, see “Generating lists in CATALOG MANAGER” on page 45.

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 on page 320) shows which privileges are affected by the REVOKE action.

**Figure 141: Cascade Report**

<table>
<thead>
<tr>
<th>DEGA-R</th>
<th>Cascade Report</th>
<th>1 to 12 of 28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command ===&gt;</td>
<td>Scroll ===&gt; PAGE</td>
<td></td>
</tr>
</tbody>
</table>

**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):** N
- **Save in SQL table:** N A/Y/R/N A/Y-Append, R-Replace
- **Name of saved SQL:** 20080918_145823

------------------------- Report of Cascaded Grants ---------------------------

<table>
<thead>
<tr>
<th>Grantee</th>
<th>Grantor</th>
<th>WithGrant</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>INSTALL SYSADM1: CSTCXN</td>
<td>SYSADM2: BMCADM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>DEM AS GRANTOR ON 2010-09-17-10.23.39.795807</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>DEM AS GRANTEE ON 2010-08-11-16.34.35.323882</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>ASSUMING DEM WAS NEVER INSTALL SYSADM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>DEM0809A DEM</td>
<td>VW</td>
<td>DEM0809A.VW04A1</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>WILL BE DROPPED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>DEM0809A DEM</td>
<td>VW</td>
<td>DEM0809A.VW0719</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>WILL BE DROPPED</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>DEM0917A DEM</td>
<td>DT</td>
<td>DAVL#1.UDOC5M</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>USAGE ON DISTINCT TYPE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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, enter 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, enter 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, enter 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, enter a name for the report.

4  Press END to display the object list panel.

**Revoking privileges on specific objects**

Use the following procedure 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 45.

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, enter **US**, and then press **Enter**. The example in **Figure 142 on page 322** shows the user privileges for a database.

**Figure 142: Object privileges panel**

```
DEFF-R ----------------------  DATABASE PRIVILEGES  --------------------------
Command ===>                                                  Scroll ===> PAGE 02
CMD will show commands for this list. Type command and press ENTER
Lists: AU
QUALIFIER: DATABASE=ACTADMN1
Cmd User           TBL TSP ADM CTL MNT DIS DRP CPY LOD REC REC REP STA STT STO
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
CATADMDB        G   G   G   G   G   G   G   G   G   G   G   G   G   G   G
CATDB1          G   G   G   G   G   G   G   G   G   G   G   G   G   G   G
CATDB1A                             Y   Y
CATDB1B                             Y   Y
CATDB1C         G   G   G   G   G   G   G   G   G   G   G   G   G   G   G
RDATLF3         G   G   G   G   G   G   G   G   G   G   G   G   G   G   G
******************************  BOTTOM OF DATA  *******************************
```

To generate a user authorizations list for that object, enter **UA**, and then press **Enter** (**Figure 143 on page 322**).

**Figure 143: User List panel**

```
DEGA-R -------------------  USER AUTHORIZATIONS LIST  ------------- ROW 1 OF 1
Command ===>                                                  Scroll ===> PAGE 02
CMD will show commands for this list. Type command and press ENTER
Lists: AU
QUALIFIER: DATABASE=ACTADMN1
Command       User      Grantor   Type HowGot  Date       Authorizations
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----
CATADMDB  CATDB1     DB  DBADM   2007-05-08 GGGGGGGGGGGGGGG-----
CATDB1    RDAPXB     DB  SYSADM  2007-05-07 GGGGGGGGGGGGGGG-----
CATDB1A   CATDB1     DB  DBMNT   2007-05-07 YY             -----      
CATDB1B   CATDB1     DB  DBMNT   2007-05-07      YY        -----      
CATDB1C   CATDB1     DB  DBADM   2007-05-08 GGGGGGGGGGGGGGG-----
RDATLF3   RDATLF3    DB          2007-04-26 GGGGGGGGGGGGGGG-----
******************************  BOTTOM OF DATA  *******************************
```

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: Confirm SQL for Revoke Reassign panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Current SQLID</th>
<th>Reassign GRANTOR</th>
<th>Edit Options</th>
<th>Reassign Grants</th>
<th>View Cascade Report</th>
<th>Save in SQL table</th>
<th>Save in PDS</th>
<th>Execute SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>RDACRJ2</td>
<td></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Current SQLID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reassign GRANTOR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit Options</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reassign Grants</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View Cascade Report</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save in SQL table</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Save in PDS</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute SQL</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name of saved SQL</td>
<td>20110121_115552</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDS(member)</td>
<td>ACT.V10.DATABASE(TEST)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute SQL</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REVOKE DBADM ON DATABASE ACTADMN1 FROM CATDB1 BY RDAPXB ;

---

5 On the Confirm SQL for Revoke Reassign panel, you can perform several actions before executing the SQL to revoke privileges:

a From the **Command** line, issue the SET sqlid command to change the value of the current SQLID.

**Note**

The ID shown in the **Current SQLID** field must have the proper authority to perform the specified SQL REVOKE statement. If you hold a primary- or secondary-authorization ID that has the proper authority, you can change the **Current SQLID** to that authorization ID and complete the CREATE. To change the **Current SQLID**, use the SET command.

b In the **Reassign GRANTOR** field, type the SQLID of a new grantor for the cascading authorizations.

**Note**

With proper authorization, you can issue the SET sqlid command on the **Command** line of the User List panel to change the value of both the **Current SQLID** and the **Reassign GRANTOR** fields.

c In the **Edit options** field, type Y to modify the default values for the options on the Confirm SQL for Revoke Reassign panel. Then, press Enter.

The Options panel is displayed. In the **Edit SQL and Confirm options** field, type Y to display the options for the Confirm SQL panel. Press END to return to the Confirm SQL for Revoke Reassign panel.

d In the **Edit SQL** field, type Y to invoke an ISPF edit session to edit the SQL. Then, press Enter.
Press END to save the SQL and return to the Confirm SQL for Revoke Reassign panel.

e In the **Reassign Grants** field, type **Y** to reassign the privileges to the SQLID specified in the **Reassign GRANTOR** field. The product will generate executable GRANT statements with the SQLID specified in the **Reassign GRANTOR** field.

**WARNING**

Revoking a privilege that created a view also drops the view and any dependent INSTEAD OF triggers. To re-create the view and the triggers, you must specify **Y** for the **Reassign Grants** field. CATALOG MANAGER then generates the CREATE VIEW and CREATE TRIGGER statements and inserts them into the SQL after the REVOKE statement.

The CREATE VIEW statement still identifies the revoked SQLID as the creator of the view. Save and edit the SQL to change the creator.

The Confirm SQL for Revoke Reassign Grants panel is displayed.

**Figure 145: Confirm SQL for Revoke Reassign Grants panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Current SQLID</th>
<th>Edit Options</th>
<th>View Cascade Report</th>
<th>Edit SQL</th>
<th>Save in SQL table</th>
<th>Name of saved SQL</th>
<th>Save in PDS</th>
<th>Execute SQL</th>
<th>SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>. . . .</td>
<td>RDACRJ2</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>N</td>
<td>20110121_115552</td>
<td>Y/AY</td>
<td>N</td>
<td>SQL</td>
</tr>
</tbody>
</table>

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.

f In the **View Cascade Report** field, enter **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 on page 325. For information about the Cascade List Report, see “Generating the cascade report” on page 319.

**Figure 146: Cascade List Report for Revoke/Reassign**

<table>
<thead>
<tr>
<th>DBDC-R</th>
<th>Cascade list report for revoke/reassign</th>
<th>1 to 11 of 12</th>
</tr>
</thead>
</table>

**Command ====>**

<table>
<thead>
<tr>
<th>Current SQLID. . . . . .</th>
<th>RDACRJ2</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Edit Options . . . . . .</th>
<th>N</th>
<th>Y/N Modify SQL processing options</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Save in SQL table. . . .</th>
<th>N</th>
<th>A/Y/R/N A/Y-Append, R-Replace</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Name of saved SQL . . .</th>
<th>20110121_115552</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Save in PDS. . . . . .</th>
<th>N</th>
<th>Y/N Save SQL in PDS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PDS(member) . . . . . .</th>
<th>ACT.V10.DATABASE(TEST)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Grantee</th>
<th>Grantor</th>
<th>WithGrant</th>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privs</td>
<td>Option</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>-----------</td>
<td>-------</td>
<td>------</td>
</tr>
</tbody>
</table>

```
-- INSTALL SYSADM1: CSTCXN   SYSADM2: RDAPXB
-- CATDB1 AS GRANTOR ON 2010-09-17-10.23.39.795807
-- CATDB1 AS GRANTEE ON 2010-08-11-16.34.35.323882
-- ASSUMING CATDB1 WAS NEVER INSTALL SYSADM

- CATADMB CATDB1    G DB ACTADMN1
- CATDB1A CATDB1    DB ACTADMN1
- CATDB1B CATDB1    CREATETAB, CREATETS DB ACTADMN1
```

**g** In the **Save in SQL table** field, enter **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_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, enter a name for the SQL.

**i** In the **Save in PDS** field, enter **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, enter the name of the PDS and member.

**k** In the **Execute SQL** field, enter **Y** to execute the SQL displayed on the Confirm SQL for Revoke Reassign panel.

**l** Press **Enter**.
The SQL Progress Indicator panel is displayed. The panel automatically refreshes to display the status of the SQL that is being executed.

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.

To verify current authorizations


   **Tip**
   To generate a cascade report, see “Generating the cascade report” on page 319.

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


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 “Revoking privileges on specific objects” on page 321.

5. To revoke the privileges of additional users, repeat Step 1 on page 326 through Step 4 on page 326.

Where to go from here

Administrators can use CATALOG MANAGER to 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 “Customizing CATALOG MANAGER command access” on page 329 to learn how to use session profile features.
Customizing CATALOG MANAGER command access

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 the following table.

Table 57: 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&lt;br&gt;Note: 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&lt;br&gt;Note: 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>

Administrators might use session profiles to:

- Remove displayed menu items that certain users do not need or want to see
- Exclude commands that certain users should be restricted from executing
- Facilitate the retrieval of DB2 objects for certain users

You can include any or all of the custom features shown in Table 57 on page 329 in a session profile.
Authorization requirements 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

You can use session profile commands to customize the Primary menu.

Use the commands that are shown in Table 58 on page 331 to customize the Primary menu and tailor the commands table when creating or editing a session profile.
Table 58: Session profile commands

<table>
<thead>
<tr>
<th>Command (short form)</th>
<th>Description</th>
</tr>
</thead>
</table>
| CUSTOMIZE (CU)       | Displays the Menu Profile Customizing panel in which you customize the CATALOG MANAGER Primary Menu. This command can be entered in the following ways:  
  - On the Command line, enter **CUSTOMIZE** `profileName` to create a new session profile or edit an existing session profile.  
  - On the Profiles List panel, enter **CU** in the Cmd column to edit an existing session profile. |
| PROFILE PROFILES (PRO) | 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. |
| SET PROFILE          | Sets the current profile to the specified profile or resets it to default values. This command can be entered in the following ways:  
  - Enter **SET PROFILE** `profileName` to set the current profile to a profile already saved.  
  - Enter **SET PROFILE OFF** to reset the profile to default values. |
| TAILOR (TA)          | 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:  
  - On the Command line, enter **TAILOR** `profileName` to create a new session profile or edit an existing session profile.  
  - On the Profiles List panel, enter **TA** in the Cmd column to edit an existing session profile. |

Creating a session profile with a customized primary menu

The following procedure describes how to create a session profile with a customized Primary Menu.

To customize the Primary Menu, you remove actions and object types from the display.
**Note**

All actions and object types shown on the default Primary Menu are always available. Users can specify them by entering their codes even if they are removed from a customized menu.

---

### To customize the Primary 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: 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  (Q) 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  C0 Column      Y  PR Proc
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: Preview of customized Primary Menu panel**

DEFF-R ---------------  Preview Menu for LI_EXAMPLE   ------------------------
Command ===>  
Select action and type object information. Then press Enter.

Action               
1. (L) List catalog objects (blank same as 0) 
2. (S) Search for catalog objects 
3. (Q) CATALOG MANAGER Options Processing 
4. (Q) List SQL for Edit and Execution 

Obj type  
11. DB Database 19. PG Package 27. LO Location  
12. SG Storgroup 20. CI Collection 28. CK Checks  
13. TB Table 21. PL Plan 29. PR Procedures  
14. VW View 22. AL Alias  
15. IX Index 24. CO Column  
16. TS Tablespace 25. DM DBRM  
18. SY Synonym 26. ST Strings  
Qualifier  
Attached to 

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.

**Displaying the session profiles list**

You can display the names of session profiles that have been created for the current SSID.

1 Display a Primary Menu panel or an object list panel.

---

*Note*  
You can check the list before creating a new session profile to ensure that you do not repeat a profile name.
On the **Command** line, type **PROFILE**, and then press **Enter**.

The Profiles List panel is displayed.

**Figure 149: Profiles List panel**

```
DEFF-R  ------------------------  Profiles List  -------------- Row 1 to 4 of 4
Command ===>                                                  Scroll ===> CSR
01
Enter CU to customize a profile menu or TA to tailor a profile command table

-- Filters --

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Profile</th>
<th>Title</th>
<th>Menu</th>
<th>Cmds</th>
<th>Db</th>
<th>Ts</th>
<th>Tb</th>
<th>Vw</th>
<th>Pl</th>
</tr>
</thead>
<tbody>
<tr>
<td>DODA</td>
<td>MIKE</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LACA</td>
<td>LACA</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LI_EXAMPLE</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MVTEST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PATTY</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

For an explanation of the columns on the Profiles List panel, press HELP.

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 333.

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: Describe Profile Entries panel**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSR 02</td>
<td></td>
</tr>
</tbody>
</table>

FROM ACT101.SEARCH_VARS

<table>
<thead>
<tr>
<th>Profile</th>
<th>LACA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>LACA</td>
</tr>
<tr>
<td>Main menu</td>
<td>Y</td>
</tr>
<tr>
<td>Command table</td>
<td></td>
</tr>
<tr>
<td>DB filter</td>
<td>AL filter</td>
</tr>
<tr>
<td>TS filter</td>
<td>XT filter</td>
</tr>
<tr>
<td>TB filter</td>
<td>SY filter</td>
</tr>
<tr>
<td>WW filter</td>
<td>SG filter</td>
</tr>
<tr>
<td>PL filter</td>
<td>SU filter</td>
</tr>
<tr>
<td>PG filter</td>
<td>US filter</td>
</tr>
<tr>
<td>IX filter</td>
<td>DN filter</td>
</tr>
<tr>
<td>CO filter</td>
<td>CK filter</td>
</tr>
</tbody>
</table>

The Describe Profile Entries panel displays the status of all types of CATALOG MANAGER customization, as follows:

- **Main menu** indicates whether the CATALOG MANAGER Primary Menu has been customized (Y or blank).

- **Command table** indicates whether the CATALOG MANAGER commands table has been tailored (Y or blank).

- The **object type** filter fields indicate whether a filter has been created for the object types shown (Y or blank).

**Creating a session profile with a tailored commands table**

Use following procedure to tailor the commands table by excluding 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.

**To tailor the commands table**

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 333), and then press Enter.

The Profile Command Tailoring List panel is displayed.

Figure 151: Section of Profile Command Tailoring List panel

```
DEFF-R --------------- Profile Command Tailoring List --- Row 1 to 16 of 257
Command ===>                                                  Scroll ===> CSR

This list is used to tailor command table for profile named LI_EXAMPLE
Enter X by a command to exclude it
Save profile variables with commands currently excluded N (Y/N)

Exclude Command      Num     Type     Comment
----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v---
ACTIVATE        00279   Cat Mgr  Make a list of aliases
AL              00002   Cat Mgr  Make a list of aliases
ALTER           00096   Cat Mgr  Generate ALTER SQL for an object
ALTER           00096   Cat Mgr  Generate ALTER SQL for an object
ALTER           00096   Cat Mgr  Generate ALTER SQL for an object
APO             00280   Cat Mgr
APPLY           00186   Cat Mgr
AU              00003   Cat Mgr  Make a list of user authorizations
AU              00003   Cat Mgr  Make a list of user authorizations
AUDIT           00088   Cat Mgr  Make a list from the audit log
BATCH           00138   Cat Mgr  Generate JCL for a CATALOG MANAGER batch
BIND            00129   Cat Mgr  Generate DSN BIND commands
BINDCOPY        00193   Cat Mgr
BINDDEPLOY      00278   Cat Mgr
BMCCHECK        00164   Cat Mgr  Generate JCL for a BMC CHECK PLUS utility
BMCHTS          00171   Cat Mgr  Generate JCL for a BMC CHECK PLUS TS utility
```

For an explanation of the columns on the Profile Command Tailoring List panel, press HELP.

2 To remove a command from the commands table, type `X` in the Exclude column beside the name of the command. You can scroll up and down to see the entire list of commands.

3 In the Save profile variables with commands currently excluded field, type `Y` to save the tailored commands table in the session profile.

4 Press Enter.

CATALOG MANAGER displays the Profile Command Tailoring List panel with the confirmation message `PROFILE. profileName ADDED`.

Creating a session profile with an initial list filter

The following procedure describes how to create an initial list filter.
You can create one initial list filter for each object type to associate with a session profile. The initial list filter applies saved search variables by default when the user with that session profile generates a list for one of these object types.

Initial list filters are very similar to saved searches. For more detailed information about creating a saved search, see “Using SEARCH to generate lists based on object attributes” on page 178.

**To create an initial list filter**

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 180.

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 BMCACT vr.SEARCH_VARS (where vr represents the CATALOG MANAGER version number).
1. Perform one of the following tasks:
   - Use the data editing feature of CATALOG MANAGER (see “Browsing and editing data” on page 215).
   - Write an interactive SQL statement.

## Editing a customized Primary Menu

Use the following procedure 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.

### To edit a customized Primary 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: Menu Profile Customizing panel to edit a session profile**

   ```
   DEFF-R ----------------- Menu Profile Customizing --------------------------
   Command ==> 
   Enter blank by items to exclude from the profile menu
   Y (L) List catalog objects (blank same as 0)
   Y (S) Search for catalog objects
   Y (C) Create objects
   Y (G) Grant privileges
   Y (O) CATALOG MANAGER options processing
   Y (D) DB2 Commands
   Y (M) Maintain logs menu
   Y (Q) List SQL for edit & execution
   Y (R) About this Release/CATALOG MANAGER Quick Reference
   Y DB Database   Y TS Tablespace   Y PL Plan   Y ST Strings
   Y SG Stogroup   Y SU SysPrivUser   Y AL Alias   Y LO Location
   Y TB Table      Y SY Synonym      Y US User    Y CK Checks
   Y VW View       Y PG Package      Y CO Column  Y PR Proc
   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.

Perform the following tasks as needed:

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**

Perform the following procedure to retrieve an initial list filter:
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 Reference Manual, Installation System Quick Start, and the BMC Products and Solutions for DB2 Customization Guide.

You can further restrict access to CATALOG MANAGER functions through plan authorizations.

Related Information

■ "Granting access to CATALOG MANAGER functions" on page 64

Determining the capabilities of a user

Several factors determine the user capabilities (regarding session profiles) that are specified in the installation options and CLIST.

The following factors should be taken into consideration:

■ 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.

1 Follow the steps in “Generating a list by using the SEARCH command” on page 180.

You can select a filter from the list to view, edit, or delete.
The BMCDB2 CLIST parameter PR, which determines which profile, if any, is invoked when a user starts CATALOG MANAGER.

Table 59 on page 341 describes how combinations of these factors determine user capabilities.

### Table 59: 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>profileName</td>
<td>When starting CATALOG MANAGER, all users on this subsystem access the session profile that is named in the PR parameter. Only users with SYSADM authority can issue the SET PROFILE profileName and SET PROFILE OFF commands.</td>
</tr>
<tr>
<td>blank</td>
<td></td>
<td>No session profile is specified in the PR parameter; therefore, no session profile is accessed by all users when CATALOG MANAGER is started. The default CATALOG MANAGER Primary Menu and commands table are available to all users, and no saved search variables are used by default for list generation. The absence of a profile name in the PR parameter means that no default session profile exists for all users. Administrators can create session profiles for specific users and assign those profiles through additional CLISTs. Only users with SYSADM authority can issue the SET PROFILE profileName and SET PROFILE OFF commands.</td>
</tr>
<tr>
<td>CRS=N</td>
<td>profileName</td>
<td>When starting CATALOG MANAGER, all users on this subsystem access the session profile that is named in the PR parameter. All users can issue the SET PROFILE profileName and SET PROFILE OFF commands.</td>
</tr>
<tr>
<td>blank</td>
<td></td>
<td>No session profile is specified in the PR parameter; therefore, no session profile is accessed by all users when CATALOG MANAGER is started. The default CATALOG MANAGER Primary Menu and commands table are available to all users, and no saved search variables are used by default for list generation. The absence of a profile name in the PR parameter means that no default session profile exists for all users. Administrators can create session profiles for specific users and assign those profiles through additional CLISTs. All users can issue the SET PROFILE profileName and SET PROFILE OFF commands.</td>
</tr>
</tbody>
</table>

### Activating and deactivating session profiles

The following procedure describes how to activate or deactivate a session profile for the current user during the current session.
You must have authorization to issue the SET PROFILE *profileName* and SET PROFILE OFF commands to perform this procedure. For more information, see “Authorization requirements to implement session profiles” on page 330.

To activate or deactivate a session profile

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 on page 342 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.

Figure 153: Customized Primary Menu panel

```
DEFF-R ----------  CATALOG MANAGER 10.01.00 Primary Menu for DATA_ENTRY

Command ===>

Select action and type object information. Then press Enter.

Action
0. (L) List using customizable lists
1. (S) Search for catalog objects
2. (O) CATALOG MANAGER Options Processing
3. (D) DB2 Commands

Obj type
11. DB Database      19. PG Package      28. CK Checks
12. SG Stogroup      20. CI Collection   29. PR Procedures
13. TB Table         21. PL Plan
14. VW View          22. AL Alias
15. IX Index         23. CO Column
16. TS Tablespace    24. ST Strings
18. SY Synonym       27. LO Location
17. SY Synonym

Qualifier
Attached to DEFF
```

3. 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 “Maintaining logs” on page 345 to learn about these helpful CATALOG MANAGER tools.
Maintaining logs

During the installation of CATALOG MANAGER, several DB2 tables are created to store various types of logged information.

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.

These tables, or logs, provide flexibility, efficiency, and integrity in reporting. User setup is minimal because logged information is captured automatically.

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.

For more information, view the Quick Course "Using Logs."

Accessing the logs

Use the following procedure to access the logs and log maintenance functions.

1. On the Primary Menu panel or any list panel, on the Command line, enter MAINTAIN ( MAINT ).
The Log Maintenance Menu panel is displayed.

**Figure 154: Log Maintenance Menu panel**

<table>
<thead>
<tr>
<th>Command =&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF-R -------------------- Log Maintenance Menu ---------------------------</td>
</tr>
<tr>
<td>Select action and log. Then press Enter.</td>
</tr>
<tr>
<td>Select . . . 1. Browse Session Log</td>
</tr>
<tr>
<td>2. Purge Session Log</td>
</tr>
<tr>
<td>4. Browse DDL Audit Log</td>
</tr>
<tr>
<td>5. Purge DDL Audit Log</td>
</tr>
<tr>
<td>7. Browse Drop Recovery Log</td>
</tr>
<tr>
<td>8. Purge Drop Recovery Log</td>
</tr>
</tbody>
</table>

Perform the following tasks as needed to browse the entries in the logs and permanently purge the logs of old entries:

---

**Note**

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.

---

**Note**

CATALOG MANAGER permanently removes all entries that are recorded before the date and time that you specify.

---

**The Session Log**

The Session Log records CATALOG MANAGER commands, DML commands, and DB2 TERM UTILITY commands executed by users during CATALOG MANAGER sessions, the data is stored until you purge or archive old entries.

Any DDL that is generated by a CATALOG MANAGER command, and then executed is automatically logged in the Audit Log.

---

**Note**
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.

1. On the Primary Menu panel or any list panel, on the **Command** line, enter **MAINTAIN** (MAINT).

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

The Browse Session Log panel is displayed.

**Figure 155: Browse Session Log panel**

```
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>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

**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: Session Log List**

<table>
<thead>
<tr>
<th>Command</th>
<th>Date</th>
<th>Authid</th>
<th>Sesn</th>
<th>Functn</th>
<th>RtnC</th>
<th>TypName</th>
</tr>
</thead>
<tbody>
<tr>
<td>--------</td>
<td>----------</td>
<td>----------</td>
<td>------</td>
<td>---------</td>
<td>------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>01</td>
<td>2010-09-20</td>
<td>RDATLF3</td>
<td>2</td>
<td>CREATE</td>
<td>000</td>
<td>TB RDATLF3.MXCDB06_TG01_</td>
</tr>
<tr>
<td>01</td>
<td>2010-09-20</td>
<td>RDATLF3</td>
<td>2</td>
<td>COMMEN</td>
<td>000</td>
<td>TB RDATLF3.MXCDB06_TG01_</td>
</tr>
<tr>
<td>01</td>
<td>2010-09-20</td>
<td>RDATLF3</td>
<td>2</td>
<td>LABEL</td>
<td>000</td>
<td>TB RDATLF3.MXCDB06_TG01_</td>
</tr>
</tbody>
</table>

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 on page 348), which shows information from the Session Log.

**Figure 157: Describe Audit Log Entry panel for a Session Log**

<table>
<thead>
<tr>
<th>Logts.</th>
<th>Authid</th>
<th>Session_id</th>
<th>Session_seq</th>
<th>Function</th>
<th>Return_code</th>
<th>Object_type</th>
<th>Object_qual</th>
<th>Object_name</th>
<th>Sequence</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010-09-20-14.57.57.933550</td>
<td>RDATLF3</td>
<td>2</td>
<td>1</td>
<td>CREATE</td>
<td>0</td>
<td>TABLE</td>
<td>RDATLF3</td>
<td>MXCDB06_TG01_</td>
<td>1</td>
<td>CREATE GLOBAL TEMPORARY TABLE RDATLF3.MXCDB06_TG01_ (COLUMN_1 CHAR(4) NOT NULL FOR SBCS DATA ,COLUMN_2 CHAR(12) NOT NULL FOR SBCS DATA ,COLUMN_3 LENG30_ABCDEFGHIJKLMZ SMALLINT NOT NULL ,COLUMN_4 INTEGER ) CCSID EBCDIC</td>
</tr>
</tbody>
</table>

9 Press END to move to the next Describe Audit Log Entry panel (if you selected multiple items) or back to the Session Log List panel.
Purging the Session Log

Because the Session Log can record almost all of the commands issued by all CATALOG MANAGER users, it can quickly grow to an unmanageable size.

The Session Log data sets can become full or grow into extents as you use CATALOG MANAGER each day. To release space, you can either purge the log to remove old entries permanently, or you can save old entries in archives.

To purge the Session Log

1. On the Primary Menu panel or any list panel, on the Command line, type MAINTAIN (MAINT).


The Purge Session Log panel is displayed.

Figure 158: Purge Session Log panel

```
DEFF-R ----------------------  Purge Session Log  ----------------------------
Command ===>
Purge will erase all rows before a specified timestamp from the Session Log. You must enter the timestamp.
Type purge specifications and press Enter.
Timestamp ........ yyyy-mm-dd.hh.mm.ss.nnnnnn
Purge now ....... N (Y/N)
```

3. In the Timestamp field, type an appropriate value, according to the format that is shown on the panel. The purge function will remove all entries that were recorded before the timestamp that you enter.

4. In the Purge now field, type Y.

   WARNING

   The purged entries cannot be retrieved.

5. Press Enter.

   After the entries are removed, CATALOG MANAGER displays the Purge Session Log panel with the message PURGE COMPLETED.
The DDL Audit Log

The DDL Audit Log is a log that 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.

Related Information

- “Browsing the Session Log” on page 347
- “Purging the Session Log” on page 349
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
- BMC Workbench

**Note**
The Drop Recovery Log does not record the DDL to recover implicitly created databases, table spaces, tables, and indexes.

For more information about dropping and recovering objects, see “Dropping and recovering objects” on page 261.

**Note**
For information about purging the Drop Recovery Log, see “Purging the Session Log” on page 349.

To browse 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: Browse Recovery Log panel**

```
DEFF-R ---------------------  Browse Recovery Log  ---------------------------
Command =>
Type browse specifications and press Enter.
Column          Operator    Value
-------------------------------------------------------------------------------
Timestamp  . . . =                                   yyyy-mm-dd-hh.mm.ss.nnnnnn
Authid . . . . . =
Object_Type  . . =
Object_Qual  . . =
Object_Name  . . =
Product  . . . . =
You may add additional conditions in the WHERE clause below
WHERE
```
3 On the Browse Recovery Log panel, in the **Authid** field, type and operator and a value for the authorization ID.

4 *(optional)* Specify a WHERE clause for the SQL statement.

If you include a WHERE clause, observe the following rules:

- Use the column names as they are displayed on the Browse Session Log panel.
- Use correct SQL case, punctuation, and syntax as required by DB2.
- Use only wildcards that are supported by DB2.

5 Press **Enter**.

The Recovery Log List panel is displayed.

**Figure 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>01</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>01</td>
<td>2011-01-27</td>
<td>16.03</td>
<td>RDAPXB2</td>
<td>AEX010</td>
<td>DB</td>
<td>PLBNAUT</td>
</tr>
<tr>
<td>01</td>
<td>2011-01-19</td>
<td>16.25</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>PLBDBA1.PLBALT02</td>
</tr>
<tr>
<td>01</td>
<td>2011-01-17</td>
<td>14.16</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>PLBDBA1.PLBALTER</td>
</tr>
<tr>
<td>01</td>
<td>2011-01-07</td>
<td>15.52</td>
<td>RDAPXB2</td>
<td>AEX010</td>
<td>DB</td>
<td>MXXCSG5.MXSS01G5</td>
</tr>
<tr>
<td>01</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>01</td>
<td>2010-11-05</td>
<td>15.52</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>QCHD14.PXBS0214</td>
</tr>
<tr>
<td>01</td>
<td>2010-09-14</td>
<td>17.22</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TB</td>
<td>QCH_LONG.QCH_LONG_NAME_TABL</td>
</tr>
<tr>
<td>01</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>01</td>
<td>2010-04-20</td>
<td>11.20</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>ACTB.CORTAPST</td>
</tr>
<tr>
<td>01</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>01</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>01</td>
<td>2010-04-19</td>
<td>15.37</td>
<td>RDAPXB2</td>
<td>ACT010</td>
<td>TS</td>
<td>ACTQX13.ACTS0113</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: Recovery Log Detail**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF-R</td>
<td></td>
<td>02</td>
</tr>
</tbody>
</table>

Enter D or S to describe a single recovery log row.

TABLESPACE=DBXNAUT.SBXNCOL

<table>
<thead>
<tr>
<th>Cmd</th>
<th>Seq</th>
<th>Type</th>
<th>Name</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TS</td>
<td>DBXNAUT.SBXNCOL</td>
<td>CREATE TABLESPACE SBXNCOL IN DB</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TB</td>
<td>RDABXN.TBXNCOL</td>
<td>CREATE TABLE RDABXN.TBXNCOL ( C</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TB</td>
<td>RDABXN.TBXNCOL</td>
<td>ALTER TABLE RDABXN.TBXNCOL ACTI</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TB</td>
<td>RDABXN.MBXN_MASK</td>
<td>CREATE MASK RDABXN.MBXN_MASK ON</td>
<td></td>
</tr>
</tbody>
</table>

8 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, which shows information from the Drop Recovery Log.

**Figure 162: Describe Audit Log Entry panel for a Drop Recovery Log**

<table>
<thead>
<tr>
<th>Command</th>
<th>Scroll</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF-R</td>
<td></td>
<td>03</td>
</tr>
</tbody>
</table>

FROM ACT101.RECOVERY_LOG

Logts: 2011-02-01-14.43.53.716869
Authid: RDAPXB2
Session_seq: 1
Sequence: 1
Drop_seq: 1
DBID: 350
OBID: 1
PSID: 2
Object_type: TABLESPACE
Object_qual: DBXNAUT
Object_name: SBXNCOL
Dep_obj_typ: TABLESPACE
Dep_obj_qal: DBXNAUT
Dep_obj_nam: SBXNCOL
Product: ACT010
Action: CREATE TABLESPACE SBXNCOL IN DBXNAUT USING STOGROUP

10 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.

**Where to go from here**

This section 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.
JCL Generation

When you generate JCL for Execution, the product 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.

**JCL Generation data sets sizing function**

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. DASD MANAGER PLUS product statistics database (statistics that the BMCSTATS utility gathers)
2. DB2 system catalog (statistics that the IBM RUNSTATS utility gathers)
3. Results of VSAM object sampling
4. Default data set allocation parameters that are set from the JCL Generation Individual Data Set Options panel

Whether it uses the BMCSTATS historical database or object sampling, or even if it does not use sizing, JCL Generation obtains some information from the DB2 system catalog.

The formulas for estimating data set size are taken from the documentation for the IBM DB2 utilities and from the documentation for the BMC utilities. Table 60 on page 356 shows the statistics that JCL Generation uses for space estimation and the source of the statistics.

<table>
<thead>
<tr>
<th>Value</th>
<th>BMCSTATS</th>
<th>IBM RUNSTATS</th>
<th>VSAM sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of active pages</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Number of modified</td>
<td>X</td>
<td>NA</td>
<td>X</td>
</tr>
<tr>
<td>pages</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Example of Worklist JCL

This is an example of batch HDDL worklist JCL.

**Figure 163: Batch HDDL worklist jcl**

```plaintext
//RDATLF4C JOB (5212), 'CATALOG',
//  CLASS=A, MSGCLASS=X, MSGLEVEL=(1,1),
//  NOTIFY=RDATLF4
//*
//* ***********************************************
//* CREATED BY : RDATLF4
//* TIMESTAMP : 10/23/2013.15.33.53
//* ENVIRONMENT: ISPF 6.3MVS TSO
//* RELEASE : V11.01.00 06/25/2013
//* DB2 VERSION: 10015
//* ***********************************************
//* -----------------------------------------------
//* CATALOG MANAGER BATCH EXECUTION -- AJX$CATB
//* -----------------------------------------------
//* CATBATCH EXEC PGM=ACTBMAIN,
//* PARM='O=ACTDOEEG,S=DEEG,I=NO,V=DEEGCAT',
//* REGION=0M
//STEPLIB DD DSN=ADM.INST1110.UDBLINK,DISP=SHR
// DD DSN=ADM.INST1110.BMCLINK,DISP=SHR
// DD DSN=RMD.INST1110.BMCLINK,DISP=SHR
// DD DSN=D2U.INST1020.BMCLINK,DISP=SHR
// DD DSN=PRF.INST1110.BMCLINK,DISP=SHR
// DD DSN=SYS3.DEEG.DSNEXIT,DISP=SHR
// DD DSN=CSGI.DB2V10M.DSNLOAD,DISP=SHR
// DD DSN=SCC.INST1110.BMCLINK,DISP=SHR
//SYSEEXEC DD DSN=ADM.INST1110.BMCREXX,DISP=SHR
```

---

**Tip**
To specify the data sizing method, see “Setting the JCL options for static data sets” on page 100.

---

<table>
<thead>
<tr>
<th>Value</th>
<th>Source</th>
<th>BMCSTATS</th>
<th>IBM RUNSTATS</th>
<th>VSAM sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page size</td>
<td></td>
<td>NA</td>
<td>X</td>
<td>NA</td>
</tr>
<tr>
<td>Maximum row length</td>
<td></td>
<td>NA</td>
<td>X</td>
<td>NA</td>
</tr>
<tr>
<td>Average row length</td>
<td>X</td>
<td>NA</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Number of rows</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Number of non-clustering indexes</td>
<td>NA</td>
<td>X</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Longest key</td>
<td>X</td>
<td>X</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Number of foreign keys</td>
<td>NA</td>
<td>X</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Number of indexes</td>
<td>X</td>
<td>X</td>
<td>NA</td>
<td></td>
</tr>
<tr>
<td>Longest foreign key</td>
<td>NA</td>
<td>X</td>
<td>NA</td>
<td></td>
</tr>
</tbody>
</table>
Example of Worklist JCL

```plaintext
//ABNLIGNR DD DUMMY
//$USEROPT DD DSN=RDATLF.SUPPORT.CAT.PROFILE(Debeopt),DISP=SHR
//ACT$MSGS DD DSN=ADM.INST1110.UDBMILIB,DISP=SHR
//SYSPRINT DD SYSOUT=*  
//ACTPRINT DD SYSOUT=*  
/** POF OVERRIDE INPUT FILE 
//AJXPOFOR DD *,DLM=$$
SRTOUT_DATACLASS=
SRTOUT_MGMTCLASS=
SRTOUT_PRIQTY=10
SRTOUT_SECQTY=2
SRTOUT_STORCLASS=
SRTOUT_UNIT=SYSDA
$$
//SYSIN DD *
DSN=RDATLF4.DEEG.HDDL3
HDDL DB ACTQX12
//*---------------------------------------------------------------
//* END OF JOBSTEP
//*---------------------------------------------------------------
//*---------------------------------------------------------------
//* END OF JOB
//*---------------------------------------------------------------
```
Using the Skeleton Library compiler

This section describes the compiler, how to test SLIBs before compiling them, and the compiler’s associated runtime unit.

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 164 on page 360 illustrates the processing flow of the SLIB compiler.
Figure 164: Processing flow of the SLIB compiler
SLIB compilation

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 “SLIB verification using ISPF file tailoring” on page 362.

SLIB changes

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.BMCCNTL 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.

---

**SLIB verification using ISPF file tailoring**

The runtime unit first attempts to process compiled SLIBs. If the runtime unit cannot process a compiled SLIB, the unit reverts to standard ISPF file tailoring.

**Note**

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 non- compilable 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)
```

**Compilation of changed SLIBs**

BMC strongly recommends that you process all SLIBs as compiled SLIBs, because the runtime performance can be adversely affected by processing non-compiled SLIBs.

**Note**

Before you test the compiled SLIB, turn off or remove any options that you used, such as $USESTFT and )CM NO-COMPILE.

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 364.

After you successfully test the SLIB using standard file tailoring, compile the SLIB into your production HLQ.UBMCLINK library.

**SLIB processing**

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 generate the SLIB report**

To use the SLIB reporting feature in Batch Execution 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:

   ```
   //JGENSRPT DD SYSOUT=*,
   // DCB=(LRECL=80,BLKSIZE=6160,RECFM=FB,DSORG=PS)
   ```

2. Resubmit your job.

   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 165 on page 364** shows a sample runtime report.

   **Figure 165: Sample runtime report**

<table>
<thead>
<tr>
<th>Skelname</th>
<th>Usage Type</th>
<th>Compile Date</th>
<th>Compile Time</th>
<th>Usage Count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>Status</td>
<td>Date</td>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>AJX#ACMX</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.41</td>
<td></td>
</tr>
<tr>
<td>AJXJOB0</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>12.28</td>
<td></td>
</tr>
<tr>
<td>AJX#USRV</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.54</td>
<td></td>
</tr>
<tr>
<td>AJXJOB5</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.52</td>
<td></td>
</tr>
<tr>
<td>AJXSTEP1</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.26</td>
<td></td>
</tr>
<tr>
<td>AJXSTEP7</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.35</td>
<td></td>
</tr>
<tr>
<td>AJXSTEPU</td>
<td>Compiled</td>
<td>03/24/2015</td>
<td>09.21</td>
<td></td>
</tr>
<tr>
<td>AJXSYSX$</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.51</td>
<td></td>
</tr>
<tr>
<td>AJXSYSMD</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>12.17</td>
<td></td>
</tr>
<tr>
<td>AJXSTWK0</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>12.14</td>
<td></td>
</tr>
<tr>
<td>AJXSYSTS</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>12.02</td>
<td></td>
</tr>
<tr>
<td>AJXISPFM</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.28</td>
<td></td>
</tr>
<tr>
<td>AJXCLI BU</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.20</td>
<td></td>
</tr>
<tr>
<td>AJXMLIBU</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.56</td>
<td></td>
</tr>
<tr>
<td>AJXSPFS</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.43</td>
<td></td>
</tr>
<tr>
<td>AJXSLIBU</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.24</td>
<td></td>
</tr>
<tr>
<td>AJXTLIBU</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.38</td>
<td></td>
</tr>
<tr>
<td>AJXPLIBU</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.29</td>
<td></td>
</tr>
<tr>
<td>AJX#PRNT</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>12.26</td>
<td></td>
</tr>
<tr>
<td>AJXNOSTS</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>12.10</td>
<td></td>
</tr>
<tr>
<td>AJXWORKK</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.31</td>
<td></td>
</tr>
<tr>
<td>AJXWORK1</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.16</td>
<td></td>
</tr>
<tr>
<td>AJWXKUNT</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>12.18</td>
<td></td>
</tr>
<tr>
<td>AJX#MTAP</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.36</td>
<td></td>
</tr>
<tr>
<td>AJXESTIM</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>12.14</td>
<td></td>
</tr>
<tr>
<td>AJX#DNS</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.42</td>
<td></td>
</tr>
<tr>
<td>AJX#SMSP</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.25</td>
<td></td>
</tr>
<tr>
<td>AJXSROH</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.23</td>
<td></td>
</tr>
<tr>
<td>AJXSORT2</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>12.15</td>
<td></td>
</tr>
<tr>
<td>AJXDISCO</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.30</td>
<td></td>
</tr>
<tr>
<td>AJXDSC1</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.59</td>
<td></td>
</tr>
<tr>
<td>AJXDISUT</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.59</td>
<td></td>
</tr>
<tr>
<td>AJXMAP0</td>
<td>Compiled</td>
<td>03/15/2015</td>
<td>11.22</td>
<td></td>
</tr>
</tbody>
</table>

Appendix B Using the Skeleton Library compiler 365
The report summary at the end of Figure 165 on page 364 provides the information shown in Table 61 on page 366.

**Table 61: Runtime report statistics**

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of FTINCLs</td>
<td>Number of file tailoring FTINCL requests</td>
</tr>
<tr>
<td>Number of )IMs</td>
<td>Number of imbeds that are encountered when FTINCLs are processed</td>
</tr>
<tr>
<td>SLIBs processed</td>
<td>Number of SLIBs</td>
</tr>
<tr>
<td>Number of JCLRECs</td>
<td>Number of JCL records</td>
</tr>
<tr>
<td>Runtime units lastcc</td>
<td>Last condition code encountered</td>
</tr>
<tr>
<td>Runtime units maxrc</td>
<td>Highest return code encountered</td>
</tr>
</tbody>
</table>
Integrating CATALOG MANAGER with the Common Explain component

This section describes how to use CATALOG MANAGER commands to navigate to the Common Explain component of the BMC SQL Explorer for DB2 and 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.

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.

Before you begin

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 Reference Manual, Installation System Quick Start, and the BMC Products and Solutions for DB2 Customization 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 166: 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 167: Using the DESCRIBE and GET commands**
- Issue the ANALYZE command to explain or edit a DML statement from the CATALOG MANAGER SQL Table.

**Figure 168: Using the ANALYZE command**

**Explaining a DBRM package or plan**

Follow this procedure to use the BMCEXPOSE 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 45.
2. In the **Command (Cm)** column beside the source object, type BMCEXPLOR (BMCEX) (as shown in Figure 169 on page 370).

**Figure 169: BMCEXPLOR command issued against plan**

<table>
<thead>
<tr>
<th>DEFF-R ---------------</th>
<th>PLAN LIST ---------------</th>
<th>Command ===》</th>
<th>Scroll ===》 CSR 01</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMD will show commands for this list. Type command and press ENTER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lists: AL CA CI DB DM DP IM IS IX MQT MX PA PG PI PL SY TB TS UA US VW LIKE %</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cm Plan     Owner   Valdat Isolat Valid Operat Acq Rel Bound      Member   Dyn</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----v----1----v----2----v----3----v----4----v----5----v----6----v----7----v----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACT102DB RDAOQL B S Y Y U C 2011-01-23</td>
<td>ACT102DG RDAOQL B S Y Y U C 2011-01-23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACT102DH RDAOQL B S Y Y U C 2011-01-23</td>
<td>ACT102DD RDAOQL B S Y Y U C 2011-01-23</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Follow this procedure to 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)

**To explain a statement from a DBRM or package**

1. Generate a list of eligible objects. For information, see “Generating lists in CATALOG MANAGER” on page 45.
2 In the Command (Cm) column beside the source object, type DESCRIBE.

3 Press Enter.

The DESCRIBE panel for the source object is displayed.

**Figure 170: DESCRIBE panel**

```
DEFF-R  Package: DSN_DEFAULT_COLLID_QZUTSTPL.QZUTSTPL    Line 1 of 42 Col 1 80
Command ===> Scroll ===> PAGE
Package = DSN_DEFAULT_COLLID_QZUTSTPL.QZUTSTPL
------------------------------------------------------------------------------
FROM SYSIBM.SYSPACKAGE
------------------------------------------------------------------------------
Location . . . .                       Collid . . . . . .._COLLID_QZUTSTPL
Name . . . . . . QZUTSTPL              Contoken . . . . 189BEEE615DFDFEE
Owner. . . . . . RDAALC                Creator. . . . . RDAALC3
Timestamp. . . . .-15-17.06.45.799274 Bindtime . . . . -15-17.15.57.045160
Qualifier. . . . QZU                   Pksize . . . . . 3752
Avgsize. . . . . 8424                  Sysentries . . . 0
Valid. . . . . . Y                     Oper . . . . . . Y
Validate . . . . R                     Isolation. . . . S
Release. . . . . C                     Explain. . . . . N
Quote. . . . . . N(')                  Comma. . . . . . N(.)
Hostlang . . . . C                     Charset. . . . . A
Mixed. . . . . . N                     Dec31. . . . . . N
Deferprep. . . . C                     SQLerror . . . . N
Remote . . . . . N                     PCtimestamp . . . ..-01-17.28.03.902847
IBMreqd. . . . . L                     Version. . . . . V6102
PDSname. . . . . RDAALC.DEV.DBRM       Degree. . . . . . 1
Group Member . .                       Dynamicrules . . B
Reoptvar . . . . N                     Deferprepare . . N
Keepdynamic. . . N                     Pathschemas. .
Type . . . . . .                       DBprotocol . . . D
Functionts . . . .-15-17.15.57.041490 Opthint. . .
Encoding CCSID . $7                   Immedwrite . . . N
Relbound . . . . O                     Catencode. . .
Remarks. . . . .                       Ownertype. . .
Rounding . . . . E                     Distribute . . . N
Lastused . . . . 01/01/0001
------------------------------------------------------------------------------
Stmtno   Stmt
24       DECLARE CRS1 CURSOR FOR
25         SELECT COLUMN_1
26           FROM QZU.QZUT01_D15S01
27           WHERE COLUMN_1 = 1234
29       OPEN CRS1
34       FETCH CRS1
35         INTO :iColumn1
38       CLOSE CRS1
------------------------------------------------------------------------------
```

4 Scroll down the DESCRIBE panel to locate the statement.
On the Command line, type GET nnn, where nnn is the number of the statement to be explained.

**Figure 171: Use of GET subcommand**

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

Stmntno Stmt
24 DECLARE CRS1 CURSOR FOR
SELECT COLUMN_1 FROM QZU.QZUT01_D15S01
WHERE COLUMN_1 = 1234
```

Press Enter.

The Confirm SQL panel is displayed.

**Figure 172: Confirm SQL panel**

```plaintext
Command ===> Confirm SQL
```

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 374.

- 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

Follow this procedure to 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)

To explain a statement from the SQL Table

1. Display the CATALOG MANAGER SQL_Table List panel. For more information, see “Displaying the SQL_Table list” on page 207.

2. In the Command (Cmd) column beside the source statement, enter ANALYZE.

   **Figure 173: SQL_Table List panel**

   ```plaintext
   DEFF-R -----------------------  SQL Table List  ---------- Row 30 to 45 of 208
   Command ===>                                                  Scroll ===> CSR
   CMD will show commands for this list. Type command and press Enter
   Subcommands are: ANALYZE, CUT, DELETE, EDIT, EXECUTE, PASTE, RENAME, TBBROWSE, TBEDIT, 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---
   MVSJXL1  DEFF04S1           SELECT COLUMN_1_LONG_COLUMN_MCIMCIMCI ,-- =
   MVSJXL1  DEFF04U1           UPDATE QCHDB01.DEFFLONG03_CREATORNAMETHATEQUA
   MVSJXL1  DEFF05I1           INSERT INTO QCHDB051.QCHSS02_TT51 WHERE ACT_
   MVSJXL1  DEFF05I2           INSERT INTO QCHDB051.QCHSS02_TT51 ( ACT_ inte
   MVSJXL1  DEFF05U1           UPDATE QCHDB051.QCHSS02_TT51 SET ACT_ SMAL
   MVSJXL1  DEFF06D1           DELETE FROM QCHDB051.QCHSS02_TT51 WHERE ACT_
   MVSJXL1  DEFF06I2           INSERT INTO QCHDB051.QCHSS02_TT51 ( ACT_INTE
   MVSJXL1  DEFF06U1           UPDATE QCHDB051.QCHSS02_TT51 SET ACT_ SMAL
   MVSJXL1  DEFF07SC           SELECT * FROM QCHDB12.QCHSP03_TT31B ; INSER
   ANALYZEVSJXL1  DEFF08SC           SELECT * FROM QCHDB12.QCHSP03_TT31B ; INSERT
   MVSJXL1  DEFF09SC           SELECT * FROM QCHDB12.DEFFLONG03_CREATORNAMET
   MVSJXL1  DEFF10SC           SELECT * FROM QCHDB12.DEFFLONG03_CREATORNAMET
   MVSJXL1  DEFF11SC           SELECT * FROM QCHDB051.QCHSS02_TT51 ; INSERT
   ```

3. Press Enter.
The Confirm SQL panel is displayed.

<table>
<thead>
<tr>
<th>Command</th>
<th>Current SQLID.</th>
<th>Edit Options</th>
<th>Save in SQL table.</th>
<th>Name of saved SQL</th>
<th>Save in PDS</th>
<th>Analysis</th>
<th>Edit/Browse data</th>
<th>Execute SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFF-R</td>
<td>RDACRJ</td>
<td>N</td>
<td>Y/N Modify SQL processing options</td>
<td>N</td>
<td>A/Y/R/N A/Y-Append, R-Replace</td>
<td>N</td>
<td>Y/N Save SQL in PDS</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20110121_115552</td>
<td></td>
<td></td>
<td>Y/N Call SQL Explorer for EXPLAIN</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>E/B/N Call the Table Editor</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y/N Execute the SQL</td>
<td>N</td>
</tr>
</tbody>
</table>

---

```sql
SELECT *
FROM QCHDBIL2.QCHSP03_TT31B
;
INSERT INTO QCHDBIL2.QCHSP03_TT31B (SSSNO, EMPNO, EMPNAME, EMDEPT).
```

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 374.

- 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

Follow this procedure to edit 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 373.

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

3. Press Enter.
An ISPF edit panel is displayed.

Figure 175: ISPF Edit panel

```plaintext
EDIT       RDACRJ.BMCCAT.WORK                              Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
MSG> -Warning- The UNDO command is not available until you change
your edit profile using the command RECOVERY ON.
000001 SELECT *  
000002 FROM QCHDBIL2.QCHSP03_TT31B  
000003 ;  
000004  
000005 INSERT INTO QCHDBIL2.QCHSP03_TT31B (  
000006 SSSNO,  
000007 EMPNO,  
000008 EMPNAME,  
000009 EMPDEPT,  
000010 EMPDEPTNAME,  
000011 EMPHIREDATE,  
000012 EMPDOB,  
000013 EMPAGE,  
000014 EMPSALARY,  
000015 EMPSEX  
000016 ) VALUES (  
000017 '11       '          , --CHAR(9)         SSSNO
```

4 Edit the SQL statement as necessary.

5 Mark the SQL text that you want to analyze by entering the block QQ command at the statement begin line and at the statement end line.

--- Note ---
SQLX does not support the use of a sequential text file that is greater than 80 bytes.

Depending on the programming language, the begin and end statements may have different keywords.

--- Note ---
If the text spans multiple panels, you might get an INVALID COMMAND message when you press DOWN because ISPF does not recognize QQ as a valid command. Ignore the INVALID COMMAND message while you continue to mark the statement, and then proceed to Step 6 on page 375 to initiate the macro. Alternatively, you can use the Q nn 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 section lists and provides descriptions of the JCL Generation keywords and variables.

AEXIN keywords

The following table lists the keywords in the AEXIN input stream.

Table 62: AEXIN keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Used by</th>
</tr>
</thead>
<tbody>
<tr>
<td>2MEGSQLE</td>
<td>Instructs the Execution component to allocate a 2-MB buffer for large SQL statements</td>
<td>E, F, J</td>
</tr>
<tr>
<td>ACM</td>
<td>Specifies the CHANGE MANAGER product</td>
<td>E</td>
</tr>
<tr>
<td>ALTERID</td>
<td>Specifies the name of the ALTER ID</td>
<td>E</td>
</tr>
<tr>
<td>ALU</td>
<td>Specifies the ALTER product</td>
<td>E</td>
</tr>
<tr>
<td>ASU</td>
<td>Specifies the DASD MANAGER PLUS product</td>
<td>E</td>
</tr>
<tr>
<td>AUC</td>
<td>Specifies the CM/PILOT component of the CHANGE MANAGER product</td>
<td>E</td>
</tr>
<tr>
<td>BINDFAIL</td>
<td>Causes worklist execution to stop with a return code of 8 if a bind fails</td>
<td>E, F, J</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>
<td></td>
</tr>
<tr>
<td></td>
<td>Without this parameter, worklist execution continues if a bind fails.</td>
<td></td>
</tr>
</tbody>
</table>
### AEXIN keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Used by</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 perform the following tasks:  
  ■ Invoke the CATALOG MANAGER product  
  ■ Override the installation option  
  ■ 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. | E       |
<p>| CATDOPT name | Provides the ALTER or CHANGE MANAGER products with the name of the installation options module for the BMC CATALOG MANAGER product                                                                                                                                          | E, F, J |</p>
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Used by</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATRECOVER</td>
<td>If the CATRECOV installation option is set to N and the keyword is in the AEXIN input stream, instructs Execution to perform the following tasks:    ■ Invoke the BMC CATALOG MANAGER product   ■ 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.</td>
<td>E, F</td>
</tr>
<tr>
<td>CATUTIL</td>
<td>For CATALOG MANAGER, specifies the worklist job</td>
<td>NA</td>
</tr>
<tr>
<td>CHECKOPT</td>
<td>Provides the BMC products with the name of the options module for the CHECK PLUS product The BMC_CHECK_OPTS keyword in the AJXPOFIN input stream replaces this keyword.</td>
<td>E</td>
</tr>
<tr>
<td>COPYOPT</td>
<td>Provides the BMC products with the name of the options module for the COPY PLUS product  The BMC_COPY_OPTS keyword in the AJXPOFIN input stream replaces this keyword.</td>
<td>E</td>
</tr>
<tr>
<td>DASDDOPT name</td>
<td>Provides ALTER or CHANGE MANAGER with the name of the installation options module for the DASD MANAGER PLUS product This keyword allows Execution to access the BMCSTATS utility.</td>
<td>E, F, J</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Used by</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>DASDTRIG</td>
<td>For DASD MANAGER PLUS, specifies running a triggered job</td>
<td>NA</td>
</tr>
</tbody>
</table>
| DB2STMSGS           | Instructs Execution to send to AEXPRINT all of the messages that are returned from DB2 when a DB2 STOP command is issued  
Usually, these messages are suppressed, and the contents are analyzed by the STOPWAIT processor. However, if the stop does not occur, you might want to view the original messages. | E       |
| DEBUGUNLD           | Enables debugging user exits, if the exits are coded to use the keyword  
This keyword also instructs Execution to turn on bit VAUNFDBG in flag field VAUNFLAG. Execution passes this bit to the user exit in the control block.                                                   | E       |
| DYNWORKUNIT         | For the Database Administration solution, defines the unit (such as SYSDA) that Execution uses to dynamically allocate temporary work data sets                                                                 | E       |
| ENV                 | Instructs the component to print ALTER or CHANGE MANAGER environment information (including a list of indexes Analysis uses) in its diagnostic output  
The ALUIN input stream also uses this keyword.                                                                 | A, B, BR, C, E, F, I, J |
| EVENTS              | For DASD MANAGER PLUS, specifies whether to record utilities in an events table                                                                                                                           | NA      |
| FLOW                | Causes Execution to produce flow trace messages in AEXPRINT when entering and exiting modules                                                                                                               | E       |
| HASHFAIL            | Causes Execution to terminate the job if a hash failure, such as a changed or added statement, occurs in a worklist                                                                                           | E, F, J |
| HASHWARNRC returnType | Defines the return code (returnCode) that Execution sends back when it finds only hash warnings  
Do not use 8 for this value.                                                                                                                         | E, F, J |

**AEXIN keywords**

---

380  **CATALOG MANAGER for DB2 User Guide**
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Used by CHANGE MANAGER component</th>
</tr>
</thead>
<tbody>
<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>
<td>E</td>
</tr>
<tr>
<td>LINES ( n )</td>
<td>Provides Execution with the number ( (n) ) of output lines per print page for AEXPRINT. The default is 56 lines per page. If you specify this keyword, you might need to change or add a /*JOBParm statement in your JCL. You can manually add or modify this keyword.</td>
<td>E</td>
</tr>
<tr>
<td>LOADOPT</td>
<td>Provides the BMC products with the name of the options module for the LOADPLUS product. The BMC_LOAD_OPTS keyword in the AJXPOFIN input stream replaces this keyword.</td>
<td>E</td>
</tr>
<tr>
<td>NEWTASKID ( a.b )</td>
<td>For CHANGE MANAGER, creates a new task ID</td>
<td>E, P</td>
</tr>
<tr>
<td>NEWWORKID ( a.b )</td>
<td>For CHANGE MANAGER, creates a new work ID. EXECUTION uses this keyword in batch mode only. The ALUIN input stream also uses this keyword.</td>
<td>E, I, P</td>
</tr>
<tr>
<td>NOAPFOK</td>
<td>Does not perform APF checking</td>
<td>E</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Used by</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| NOFAILNOIMAGECPY        | For ALTER, CHANGE MANAGER, BMC Workbench, and CATALOG MANAGER, instructs Execution to invoke the Drop Recovery feature of the CATALOG MANAGER product and to allow an object to be dropped when the following conditions exist:  
  - No image copies of the object exist.  
  - The CATRECOVER keyword is specified in the AEXIN input stream.  
  The product automatically generates the NOFAILNOIMAGECPY keyword in the AEXIN input stream when one of the following conditions exists:  
  - The DROPR_NOIC POF keyword is set to Y.  
  - The NOFAILNOIMAGECPY option is set to Y on the JCL Generation Debugging, Display and Execution panel.  
  This keyword is useful only if you have installed CATALOG MANAGER.  
  This keyword overrides the DROPR_NOIC keyword in the AJXPOFIN input stream.                                                                 | E, J     |
| NOLOADCOMP              | Instructs Execution not to compress extra spaces out of LOAD statements                                                                                                                                     | E       |
| NOSQLCOMP               | Instructs Execution not to compress extra spaces out of SQL statements                                                                                                                                 | E       |
| NOSTARTOVER             | Instructs Execution not to start the worklist again from the beginning of the worklist  
  See also the STARTOVER keyword in this table.                                                                                                                                                              | E, P     |
| NOTIFYUNLD n            | Instructs Execution to send a message to AEXPRINT every n records during an unload                                                                                                                           | E       |
| NOWKIDREPLACE           | For CHANGE MANAGER, instructs Import not to replace the changes in an existing work ID with an imported file                                                                                        | E, P     |
| REBINDFAIL              | Causes worklist execution to stop with return code 8 if a rebind fails  
  The stop is noted in the sync tables, and an Execution restart continues with the command that caused the failure.  
  Without this parameter, worklist execution continues if a rebind fails.                                                                                                                                  | E, F, J  |
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Used by CHANGE MANAGER component</th>
</tr>
</thead>
<tbody>
<tr>
<td>REBINDRC $n$</td>
<td>Allows worklist execution to continue if a rebinding fails, but returns the $n$ value for a final condition code instead of 4, the default value for the final condition code. When running standard JCL, the condition code is added to the step subsequent to the REBIND step. Execution writes warning messages to AEXPRINT but does not post entries in the sync tables.</td>
<td>E, F, J</td>
</tr>
<tr>
<td>RECOVEROPT</td>
<td>Provides the BMC products with the name of the options module for the RECOVER PLUS product. The BMC_RECOVER_OPTS keyword in the AJXPOFIN input stream replaces this keyword.</td>
<td>E</td>
</tr>
<tr>
<td>REORGOPT</td>
<td>Provides the BMC products with the name of the options module for the REORG PLUS product. The BMC_REORG_OPTS keyword in the AJXPOFIN input stream replaces this keyword.</td>
<td>E</td>
</tr>
<tr>
<td>REPLACETASKID $a.b$</td>
<td>For CHANGE MANAGER, replaces an existing task ID</td>
<td>E, P</td>
</tr>
<tr>
<td>REPLACEWORKID $a.b$</td>
<td>For CHANGE MANAGER, replaces an existing work ID. The ALUIN input stream also uses this keyword.</td>
<td>E, F, I, P</td>
</tr>
<tr>
<td>RESTART</td>
<td>Instructs Execution to restart a worklist from the last sync or stop point. RESTART fails if no -STOP command or error sync point (-SYNC) exists in the worklist. You cannot specify the RESTART keyword with the STARTOVER keyword.</td>
<td>E, J</td>
</tr>
<tr>
<td>RESTARTTPARM $parameter$</td>
<td>During Execution restart, passes a user-defined parameter string ($parameter$) to the utility that is being restarted. The form of the parameter string depends on the utility that is being restarted.</td>
<td>E, J</td>
</tr>
<tr>
<td>SPBXPRINT</td>
<td>Prints the output from a stored procedure</td>
<td>E</td>
</tr>
<tr>
<td>SSID $ssid$</td>
<td>Identifies the DB2 subsystem ID or the DB2 data sharing group attachment name. The SSID keyword must match the -SSID command in the worklist. This parameter is required. If the SSID is specified in the JCL in an EXEC statement in a cataloged procedure (which begins with a PROC statement), its value overrides the value of the SSID keyword. The AJXIN and ALUIN input streams also use this keyword.</td>
<td>A, B, BR, E, F, I, J, P, S</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Used by CHANGE MANAGER component</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------</td>
</tr>
</tbody>
</table>
| STARTOVER | Instructs Execution to start the worklist again from the beginning of the worklist  
You cannot specify the STARTOVER keyword with the RESTART keyword. See also the NOSTARTOVER keyword in this table.                                                    | E, J, P                           |
| STATS     | Prints the execution statistics                                                                                                                                                                              | E                                |
### AEXIN keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Used by</th>
</tr>
</thead>
</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). | E, F, J               |
<p>| 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. | E, F, J               |</p>
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Used by</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNDELETE</td>
<td>Instructs Execution to remove all sync entries when an Execution job completes with no errors</td>
<td>E, F, J</td>
</tr>
<tr>
<td>SYNLIST</td>
<td>Prints a synonym list</td>
<td>E</td>
</tr>
<tr>
<td>TASKID \textit{a.b}</td>
<td>For CHANGE MANAGER, specifies the task ID to use</td>
<td>E, P</td>
</tr>
<tr>
<td>UNLOADDOPT</td>
<td>Specifies the name of the options module for the UNLOAD PLUS product</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>The BMC_UNLOAD_OPTS keyword in the AJXPOFIN input stream replaces this keyword.</td>
<td></td>
</tr>
<tr>
<td>UTILITYID</td>
<td>For DASD MANAGER PLUS, specifies the utility ID to use for the utilities</td>
<td>NA</td>
</tr>
<tr>
<td>VCAT</td>
<td>For DASD MANAGER PLUS, identifies the VCAT for jobs</td>
<td>NA</td>
</tr>
<tr>
<td></td>
<td>The AJXIN input stream also uses this keyword.</td>
<td></td>
</tr>
<tr>
<td>WARNRC</td>
<td>Specifies the return code to use for warnings</td>
<td>E</td>
</tr>
<tr>
<td>WORKID \textit{a. b}</td>
<td>Specifies the work ID to use</td>
<td>A, B, E, I</td>
</tr>
<tr>
<td></td>
<td>Execution fails if this work ID does not match the work ID that the \textit{-WKID} command in the worklist specifies. The AJXIN and ALUIN input streams also use this keyword.</td>
<td></td>
</tr>
</tbody>
</table>

### Symbolic variables for BMC Administrative products

You can use symbolic variables in the installation options module, 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 63 on page 387 lists all of the symbolic variables that the JCL Generation and Execution components use. 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 information about the use of symbolic variables, see the

Table 63: Symbolic variables for BMC Software Administrative products

<table>
<thead>
<tr>
<th>Symbolic variable and description</th>
<th>Size</th>
<th>Value</th>
<th>Related SLIB variable</th>
<th>OUTPUT descriptor variable</th>
<th>TEMPLATE descriptor variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJXODS44</td>
<td>1</td>
<td>Y N</td>
<td>AJXDSN44</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Whether JCL Generation verifies that the prefix of a data set name contains 44 characters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALID a b</td>
<td>8</td>
<td>ALTER CATMGR DASDMGR</td>
<td>AJXJAID</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>Product ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATTACH</td>
<td>4</td>
<td>None</td>
<td>AJXJSSID</td>
<td>ATTACH</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>DB2 subsystem ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BMCCP</td>
<td>2 to 4</td>
<td>None</td>
<td>AJXBMCCP</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>Partition number in which insignificant digits are suppressed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CR a b</td>
<td>8</td>
<td>None</td>
<td>AJXCR</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>Creator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DA a b</td>
<td>2</td>
<td>DD</td>
<td>AJXYMD</td>
<td>DAY</td>
<td>DA DAY</td>
</tr>
<tr>
<td>DAY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day part of YYMMDD format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATE a b</td>
<td>6</td>
<td>YYMMDD</td>
<td>AJXYMD</td>
<td>DATE</td>
<td>DATE</td>
</tr>
<tr>
<td>System date (same format as JYMD and YMD variables)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATEJ</td>
<td>7</td>
<td>YYYYDDD</td>
<td>AJX4YDDD</td>
<td>JDATE</td>
<td>JDATE</td>
</tr>
<tr>
<td>Julian date shown with four-digit year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbolic variable and description</td>
<td>Size</td>
<td>Value</td>
<td>Related SLIB variable</td>
<td>OUTPUT descriptor variable</td>
<td>TEMPLATE descriptor variable</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------</td>
<td>------------------------</td>
<td>-----------------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>DB</td>
<td>8</td>
<td>Name of current database</td>
<td>AJXDB</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>DBNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DB2V2</td>
<td>2</td>
<td>Version of DB2</td>
<td>AJXDB2V2</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>DB2V3</td>
<td>3</td>
<td>Version of DB2</td>
<td>AJXDB2V3</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>DDD</td>
<td>3</td>
<td><em>DDD</em></td>
<td>AJXYYDDD</td>
<td>JDAY</td>
<td>JDAY</td>
</tr>
<tr>
<td>Julian day</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDNAME</td>
<td>8</td>
<td>None</td>
<td>AJXJDDN</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>DDname</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDOPT</td>
<td>8</td>
<td>None</td>
<td>AJXDDOPT</td>
<td>SEQ</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>Name of the installation options module</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>GDG</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>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>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 = $Local</td>
<td>AJXIC</td>
<td>ICTYPE</td>
<td>IC ICTYPE</td>
</tr>
<tr>
<td>ICTYPE</td>
<td></td>
<td>$R = $Remote</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image copy type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IS</td>
<td>8</td>
<td>None</td>
<td>AJXIXSPC</td>
<td>TS</td>
<td>TS</td>
</tr>
<tr>
<td>Index space name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IX</td>
<td>16</td>
<td>Name of current index</td>
<td>AJXIX</td>
<td>TS</td>
<td>IS</td>
</tr>
<tr>
<td>IXNAME</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IXCR</td>
<td>8</td>
<td>Creator of current index</td>
<td>AJXCR</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>Index creator name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IXNODE</td>
<td>22</td>
<td>None</td>
<td>AJXIXNOD</td>
<td>DB.TS</td>
<td>DB.IS</td>
</tr>
<tr>
<td>Index node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IXSPC</td>
<td>8</td>
<td>None</td>
<td>AJXIXSPC</td>
<td>TS</td>
<td>IS</td>
</tr>
<tr>
<td>Index space name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JD</td>
<td>3</td>
<td>$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>
</tbody>
</table>
## Symbolic variables for BMC Administrative products

<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>
| **JOBCHAR** a b **Component for which JCL is being generated** | 1    | A = Analysis  
B = Baseline  
C = Compare  
E = Execution  
I = Import | AJXFJCHR | JOBNAME | JOBNAME |
| **JOBNAME** Work ID or name assigned to a job | 8    | Work ID or job name in the JOB statement | AJXWKID AJXJOBNM | JOBNAME | JOBNAME |
| **JOBTYP** a b **Type of job** | 8    | ANALYSIS EXECUTION BASELINE COMPARE IMPORT CATMGR UTILITY DEFINE CAT ALTER CHGMGR DASDMGR | AJXJOBT | TYPE | IC |
| **JPCOD** a **Product code** | 3    | ACM = CHANGE MANAGER  
ACT = CATALOG MANAGER  
ALU = ALTER  
ASU = DASD MANAGER PLUS | AJXJPCOD | JOBNAME | JOBNAME |
<p>| <strong>JS1</strong> Job sequence number | 1    | None | None | None | None |
| <strong>JS2</strong> Job sequence number | 2    | None | None | None | None |
| <strong>JS4</strong> Job sequence number | 4    | None | None | None | None |
| <strong>JQID</strong> Work ID | 8    | Work ID | AJXJQID | UTIL | UT |</p>
<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>JSSID e</td>
<td>4</td>
<td>None</td>
<td>AJXSSID</td>
<td>SSID</td>
<td>SSID</td>
</tr>
<tr>
<td>DB2 subsystem ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JU</td>
<td>7</td>
<td>YYYYYDDD</td>
<td>AJX4YDDDD</td>
<td>JDATE</td>
<td>JDATE</td>
</tr>
<tr>
<td>Julian date shown with four-digit year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JUL4Y a</td>
<td>7</td>
<td>YYYYYDDD</td>
<td>AJX4YDDDD</td>
<td>JDATE</td>
<td>JDATE</td>
</tr>
<tr>
<td>Julian date</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JULIAN a b</td>
<td>5</td>
<td>YYDDD</td>
<td>AJXYYYDDDD</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</td>
<td>AJXLR</td>
<td>TYPE</td>
<td>LOCREM</td>
</tr>
<tr>
<td>Image copy type</td>
<td></td>
<td>R = Remote</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</td>
<td>AJXLR</td>
<td>TYPE</td>
<td>LR</td>
</tr>
<tr>
<td>Image copy type</td>
<td></td>
<td>R = Remote</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbolic variable and description</td>
<td>Size</td>
<td>Value</td>
<td>Related SLIB variable</td>
<td>OUTPUT descriptor variable</td>
<td>TEMPLATE descriptor variable</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------</td>
<td>-------</td>
<td>------------------------</td>
<td>-----------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>MEMBER a b</td>
<td>8</td>
<td>First seven characters of the work ID, with one of the following values appended: J = JCL W = Worklist D = Diagnostics C = CDL S = Source (CM/PILOT) T = Target (CM/PILOT) Z = Other Note: If the work ID template is used in CM/PILOT, then the task ID is used.</td>
<td>AJXMEMBER</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>MEMBR PDS member name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDS member name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MI</td>
<td>2</td>
<td>MM</td>
<td>AJXHMS</td>
<td>MINUTE</td>
<td>MI MINUTE</td>
</tr>
<tr>
<td>MINUTE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minute part of HHMMSS format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MMDD</td>
<td>4</td>
<td>MMDD</td>
<td>MONTH.DAY</td>
<td>MONTH.DAY</td>
<td></td>
</tr>
<tr>
<td>JCL date</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MO</td>
<td>2</td>
<td>MM</td>
<td>AJXYMD</td>
<td>MONTH</td>
<td>MO MONTH</td>
</tr>
<tr>
<td>MONTH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Month part of YYMMDD format</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSSID</td>
<td>4</td>
<td>Determined at runtime from the -JCLP command in the worklist</td>
<td>AJXMSSID</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>OBJT</td>
<td>2</td>
<td>Object type: TS, TB, IX</td>
<td>AJXOBT</td>
<td>UTIL TYPE</td>
<td>IC</td>
</tr>
<tr>
<td>OBJTYP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OBNAM</td>
<td>27</td>
<td>None</td>
<td>AJXOBNAM</td>
<td>DB.TS CREATOR.IX</td>
<td>DB..SN</td>
</tr>
<tr>
<td>Object name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbolic variable and description</td>
<td>Size</td>
<td>Value</td>
<td>Related SLIB variable</td>
<td>OUTPUT descriptor variable</td>
<td>TEMPLATE descriptor variable</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------</td>
<td>-------</td>
<td>------------------------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>OBNOD Database and either table space name or index space name</td>
<td>17</td>
<td>#### (for objects that do not exist)</td>
<td>AJXOBNOD</td>
<td>DB.TS</td>
<td>DB..SN</td>
</tr>
<tr>
<td>PA Partition number</td>
<td>3</td>
<td>None</td>
<td>AJXPARTC</td>
<td>PART</td>
<td>PA</td>
</tr>
<tr>
<td>PART Partition number in which insignificant digits are suppressed</td>
<td>4</td>
<td>None</td>
<td>AJXPARTC</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>PART4 Partition number in which leading zeros are not suppressed</td>
<td>4</td>
<td>None</td>
<td>AJX4PART</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>PART5 Partition number in which leading zeros are not suppressed</td>
<td>5</td>
<td>None</td>
<td>AJX5PART</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>PB PRIBAC Type of copy</td>
<td>1</td>
<td>P = Primary B = Backup</td>
<td>AJXPB</td>
<td>TYPE</td>
<td>PB PRIBAC</td>
</tr>
<tr>
<td>PGMR Programmer name field</td>
<td>20</td>
<td>‘&amp;&amp;JOBTYP - &amp;&amp;WKID’</td>
<td>AJXPGMR</td>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>PREFIX (ALTER and CHANGE MANAGER) TSO prefix (same as ZPREFIX)</td>
<td>8</td>
<td>TSO prefix (ZUSER if NOPREFIX) or user ID</td>
<td>ZPREFIX</td>
<td>None</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>RHLQ High-level qualifier</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# Restart sequence number</td>
<td>6</td>
<td>None</td>
<td>AJXRSEQ#</td>
<td>SEQ</td>
<td>SEQ</td>
</tr>
<tr>
<td>RTYPE Object type</td>
<td>2</td>
<td>None</td>
<td>AJXOBJT</td>
<td>TYPE</td>
<td>IC</td>
</tr>
<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>RUNTYP a b</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 SC SECOND</td>
</tr>
<tr>
<td>SEC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECOND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECOND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECOND</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>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>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>AJXJSSID</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 a b</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>
<tr>
<td>STEP#</td>
<td>6</td>
<td>None</td>
<td>AJXSTEEPC</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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbolic variable and description</td>
<td>Size</td>
<td>Value</td>
<td>Related SLIB variable</td>
<td>OUTPUT descriptor variable</td>
<td>TEMPLATE descriptor variable</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>------</td>
<td>-------</td>
<td>-----------------------</td>
<td>---------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>SYSMLIB System ISPF message library</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 JCL symbolic parameter</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>TBCR TBCRE Table creator name</td>
<td>8</td>
<td>Owner of current table</td>
<td>AJXTBCR AJXTBCRE</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>TBNAM d TBNNAME Table name</td>
<td>12</td>
<td>Name of current table</td>
<td>AJXTBNAM</td>
<td>TS</td>
<td>TS</td>
</tr>
<tr>
<td>TBNODE Table node</td>
<td>22</td>
<td>None</td>
<td>AJXTBNOD</td>
<td>DB.TS</td>
<td>DB..TS</td>
</tr>
<tr>
<td>TI TIME a b System time</td>
<td>6</td>
<td>HHMMSS</td>
<td>AJXHMS</td>
<td>TIME</td>
<td>TI</td>
</tr>
<tr>
<td>TIME4 Hours and minutes of HHMMSS format</td>
<td>4</td>
<td>HHMM</td>
<td>AJXHMS</td>
<td>HOUR.MINUTE</td>
<td>HO.MI</td>
</tr>
<tr>
<td>TS TSNNAME Table space name</td>
<td>8</td>
<td>Name of current table space</td>
<td>AJXTS</td>
<td>TS</td>
<td>TS</td>
</tr>
<tr>
<td>TSCR Table space creator</td>
<td>8</td>
<td>None</td>
<td>AJXTSCR</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>TSIX 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>
<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>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>TU3</td>
<td>4</td>
<td>None</td>
<td>AJXTU3</td>
<td>STEPNAME</td>
<td>STEPNAME</td>
</tr>
<tr>
<td>Tape unit 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYPE</td>
<td>1</td>
<td>L = Local</td>
<td>AJXLR</td>
<td>TYPE</td>
<td>IC</td>
</tr>
<tr>
<td>Type of copy</td>
<td></td>
<td>R = Remote</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UCMD</td>
<td>4</td>
<td>None</td>
<td>AJXUCMD</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>Command ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UDOPT</td>
<td>8</td>
<td>None</td>
<td>AJXUDOPT</td>
<td>STEPNAME</td>
<td>STEPNAME</td>
</tr>
<tr>
<td>Installation options file name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UID</td>
<td>7</td>
<td>None</td>
<td>ZUSER</td>
<td>None</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>TSO user ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ULLQ</td>
<td>4</td>
<td>Low-level qualifier for user-defined data sets</td>
<td>AJXULLQ</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Low-level qualifier for user-defined data sets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP</td>
<td>3</td>
<td>None</td>
<td>AJXUPART</td>
<td>PART</td>
<td>PART</td>
</tr>
<tr>
<td>UPART</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partition number variable compared to AJXPARTC in which insignificant digits are suppressed For example, if a partition number is 10, AJXUPART will contain 010, while AJXPARTC will contain 10.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER1</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>User-defined</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USER2</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>User-defined</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USERID a b</td>
<td>7</td>
<td>TSO user ID</td>
<td>ZUSER</td>
<td>USERID</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>TSO user ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UT</td>
<td>16</td>
<td>None</td>
<td>AJXUTID</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>UTID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UTILID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Utility ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

CATALOG MANAGER for DB2 User Guide
<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>UTILPFX First eight bytes of utility ID</td>
<td>8</td>
<td>None</td>
<td>AJXUTID</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>UTILSFX Last eight bytes of utility ID</td>
<td>8</td>
<td>None</td>
<td>AJXUTID</td>
<td>UTIL</td>
<td>UT</td>
</tr>
<tr>
<td>UVR1 UVR2 UVR3 UVR4 UVR5 User-defined character variable</td>
<td>8</td>
<td>User-defined variable or # repeated 8 times (if value is blank)</td>
<td>AJXUVR1 AJXUVR2 AJXUVR3 AJXUVR4 AJXUVR5</td>
<td>User-defined</td>
<td>User-defined</td>
</tr>
<tr>
<td>VCAT VCAT name</td>
<td>8</td>
<td>None</td>
<td>AJXVCAT</td>
<td>DB</td>
<td>DB</td>
</tr>
<tr>
<td>(DASD MANAGER PLUS)WKID Name of the work ID</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>WKOWN WKOWNER Work ID owner</td>
<td>8</td>
<td>Owner of the current work ID in use</td>
<td>AJXWKOWN</td>
<td>JOBNAME STEPNAME</td>
<td>JOBNAME STEPNAME</td>
</tr>
<tr>
<td>WORKID Work ID name</td>
<td>18</td>
<td>Name of the current work ID in use or, for Compare only, the outbound migrate profile name specified that is for use If the work ID name contains characters that are invalid for use in data set names, the work ID will be truncated at the first invalid character.</td>
<td>AJXJQID</td>
<td>UTIL</td>
<td>UT</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>WORKID8 a b</td>
<td>8</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>AJXWKID</td>
<td>UT</td>
</tr>
<tr>
<td>First eight characters of the work ID name</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YE</td>
<td>4</td>
<td>YYYY</td>
<td>AJX4YDDD</td>
<td>YEAR</td>
<td>YY</td>
</tr>
<tr>
<td>YEAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year from Julian date</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YMD</td>
<td>6</td>
<td>YYMMDD</td>
<td>AJXYMD</td>
<td>DATE</td>
<td>DATE</td>
</tr>
<tr>
<td>Date of JCL creation (same as DATE and JYMD)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YY</td>
<td>2</td>
<td>YY</td>
<td>AJXYYDDDD</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Year from Julian date</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YYDDD</td>
<td>5</td>
<td>YYDDD</td>
<td>AJXYYDDDD</td>
<td>JDATE</td>
<td>JDATE</td>
</tr>
<tr>
<td>Julian date of JCL creation (same as JULIAN)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YYYYYDDD</td>
<td>7</td>
<td>YYYYYDDD</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>ZACCTNUM</td>
<td>40</td>
<td></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>
</tr>
<tr>
<td>User’s account number for jobs that are generated by the product</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZPREFIX a b g</td>
<td>8</td>
<td>None</td>
<td>ZPREFIX</td>
<td>USERID</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>TSO prefix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZSYSID</td>
<td></td>
<td>System name</td>
<td>ZSYSID</td>
<td>ATTACH</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>ISPF system variable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZUSER a b</td>
<td>8</td>
<td>None</td>
<td>ZUSER</td>
<td>USERID</td>
<td><strong>PREFIX</strong></td>
</tr>
<tr>
<td>User ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbolic variable and description</td>
<td>Size</td>
<td>Value</td>
<td>Related SLIB variable</td>
<td>OUTPUT descriptor variable</td>
<td>TEMPLATE descriptor variable</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>------</td>
<td>-------</td>
<td>-----------------------</td>
<td>---------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td><strong>a</strong> ALTER and CHANGE MANAGER resolve this variable for job cards and data set names.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>b</strong> DASD MANAGER PLUS resolves this variable for job cards and data set names.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>c</strong> 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:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ JCL Generation does not verify that the GDG base definitions already exist.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>■ No GDG numbers are built for invalid or incomplete table space names. JCL Generation builds the &amp;AJXDB, &amp;AJXTS, and &amp;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.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>d</strong> This value truncates after eight characters when used by JCL Generation.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>e</strong> This value is determined at runtime (same as SSID).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>f</strong> For the Compare component of CHANGE MANAGER, this variable indicates the outbound migrate profile name.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>g</strong> Leave this variable blank for NOPREFIX (same as PREFIX).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CATALOG MANAGER installation options

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

Installation option example

The following figure provides an example of the installation options module for CATALOG MANAGER.

Figure 176: CATALOG MANAGER installation options module

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

******************************************************************************
* 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_DMAIN,*
ICCOL=,*
ICSYC=,*
CATOP=Y,*
PDSN=(&&ZUSER..BMCCAT.PRINT,R),*
WDSN=(&&ZUSER..BMCCAT.WORK,R),*
ADSN=(&&ZUSER..BMCCAT.ARCHIVE,R),*
BDSN=('''BMCADMN.V111.D10.OBDBRM''',R),*
LDSN=(&&ZUSER..BMCCAT.SQL,R),*
JDSN=(&&ZUSER..BMCCAT.JCL(),R),*
UODSN=('''&&ZUSER..BMCCAT.USEROPT''',R),*
TDSN=(,R),*
PODFS=('''BMCADMN.D10.UDBCNTL(AJXB1PDF)'',R),*
HDT5=Y,*
HDTB=Y,*
HDAL=N,*
HDIX=Y,*
HDSY=N,*
HDWV=Y,*
HDPL=Y,*
HDRT=Y,*
TNCC=<>,*
TNLMR=M,*
AOPTS=(ACMDOPD1,R),*
BOPTS=(ASUDDPDI,R),*
GPLAN=ACT911DG*
XODSN=('''BMCADMN.V10.STDCUST.DBXML''',R)
$ACTSQLD AUDPOL=DYNAMIC,*
AUTOALE=DYNAMIC,*
AUTORUN=DYNAMIC,*
AUTOWIN=DYNAMIC,*
AUXRELS=DYNAMIC,*
CHECKDE=DYNAMIC,*
CHECKS=DYNAMIC,*
CHECKS2=DYNAMIC,*
COLAUTH=DYNAMIC,*
COLDISH=DYNAMIC,*
COLDISS=DYNAMIC,*
COLDIST=DYNAMIC,*
COLSTAT=DYNAMIC,*
COLUMNH=DYNAMIC,*
COLUMNS=DYNAMIC,*
CONSTDE=DYNAMIC,*
CONTDL=DYNAMIC,*
CONTXT=DYNAMIC,*
COPY=DYNAMIC,*
CXATTR=DYNAMIC,*
CXAUTH=DYNAMIC,*
DATABASE=DYNAMIC,*
DATATYPE=DYNAMIC,*
DBAUTH=DYNAMIC,*
DBRM=DYNAMIC,*
DEPEND=DYNAMIC,*
ENVIRON=DYNAMIC,*
FIELDS=DYNAMIC,*
FOREIGN=DYNAMIC,*
INDEXES=DYNAMIC,*
INDEXH=DYNAMIC,*
INDEXPA=DYNAMIC,*
INDEXPH=DYNAMIC,*
INDEXSH=DYNAMIC,
Installation option example
Installation option descriptions

This section describes the installation options and, in some cases, indicates the default value of the option.

**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.

### Descriptions of the $ACTDOPT options

#### 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 CATALOG MANAGER installation options module name to be 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.

#### AUDIT=Y

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

#### BDSN="&&HLQ..BMCDBRM"

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

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).

HDT$=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=ACTvrDH

This option specifies the plan for displaying BMC utility status.

ICCOL

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

ICSYC

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=ACTvrDK

This option specifies the name of the DB2 commands plan.
**LDSN=&&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:

```plaintext
DCB=(LRECL=4092,BLKSIZE=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=&&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:

```plaintext
DCB=(LRECL=4092,BLKSIZE=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..UBMCCNTL(&POFNAME)'**

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

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

SPLAN=ACTvrDSdt

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

TDSNdt

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=biggersmallersymbols

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:

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

TRS=N

This option indicates whether all users or just users with DB2 SYSADM authority can terminate utilities.

<table>
<thead>
<tr>
<th>Y</th>
<th>Only users with DB2 SYSADM authority can terminate utilities.</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Any user can terminate the utility.</td>
</tr>
</tbody>
</table>

UCOMD

(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:BMCNCTL 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.

**VAR=DYNAMIC**

This option specifies that CATALOG MANAGER uses dynamic SQL to access the SYSIBM.SYSVARIABLES 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 this option.

**VARAUTH=DYNAMIC**

This option specifies that CATALOG MANAGER uses dynamic SQL to access the SYSIBM.SYSVARIABLEAUTH 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 this option.

**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.

**Descriptions of the $ACTSQLD options**

The $ACTSQLD options that are listed in “Installation option example” on page 401 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 64 on page 411 lists the $ACTSQLD options and the corresponding table that CATALOG MANAGER accesses.

**Table 64: $ACTSQLD options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Catalog table</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUDPOL</td>
<td>SYSIBM.SYSAUDITPOLICIES</td>
</tr>
<tr>
<td>AUTOALE</td>
<td>SYSIBM.SYSAUTOALERTS</td>
</tr>
<tr>
<td>AUTORUN</td>
<td>SYSIBM.SYSAUTORUNS_HIST</td>
</tr>
<tr>
<td>AUTOWIN</td>
<td>SYSIBM.SYSAUTOTIMEWINDOWS</td>
</tr>
<tr>
<td>AUXRELS</td>
<td>SYSIBM.SYSAUXRELS</td>
</tr>
<tr>
<td>CHECKDE</td>
<td>SYSIBM.SYSCHECKDEP</td>
</tr>
<tr>
<td>CHECKS</td>
<td>SYSIBM.SYSCHECKS</td>
</tr>
<tr>
<td>CHECKS2</td>
<td>SYSIBM.SYSCHECKS2</td>
</tr>
<tr>
<td>COLAUTH</td>
<td>SYSIBM.SYSCOLAUTH</td>
</tr>
<tr>
<td>COLDISH</td>
<td>SYSIBM.SYSCOLDIST_HIST</td>
</tr>
<tr>
<td>COLDISS</td>
<td>SYSIBM.SYSCOLDISTSTATS</td>
</tr>
<tr>
<td>COLDIST</td>
<td>SYSIBM.SYSCOLDIST</td>
</tr>
<tr>
<td>COLSTAT</td>
<td>SYSIBM.SYSCOLSTATS</td>
</tr>
<tr>
<td>COLUMNSH</td>
<td>SYSIBM.SYSCOLUMNS_HIST</td>
</tr>
<tr>
<td>COLUMNS</td>
<td>SYSIBM.SYSCOLUMNS</td>
</tr>
<tr>
<td>CONSTDE</td>
<td>SYSIBM.SYSCONSTDEP</td>
</tr>
<tr>
<td>CONTRL</td>
<td>SYSIBM.SYSControlS</td>
</tr>
<tr>
<td>CONTEXT</td>
<td>SYSIBM.SYSCONTEXT</td>
</tr>
<tr>
<td>COPY</td>
<td>SYSIBM.SYSCOPY</td>
</tr>
<tr>
<td>CXATTR</td>
<td>SYSIBM.SYSCXTXTTRUSTATTRS</td>
</tr>
<tr>
<td>Option</td>
<td>Catalog table</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>CXAUTH</td>
<td>SYSIBM.SYSCONTEXTAUTHIDS</td>
</tr>
<tr>
<td>DATABAS</td>
<td>SYSIBM.SYSDATABASE</td>
</tr>
<tr>
<td>DATATYP</td>
<td>SYSIBM.SYSDATATYPES</td>
</tr>
<tr>
<td>DBAUTH</td>
<td>SYSIBM.SYSDBAUTH</td>
</tr>
<tr>
<td>DBRM</td>
<td>SYSIBM.SYSDBRM</td>
</tr>
<tr>
<td>DEPEND</td>
<td>SYSIBM.SYSDEPENDENCIES</td>
</tr>
<tr>
<td>ENVIRON</td>
<td>SYSIBM.SYSENVIRONMENT</td>
</tr>
<tr>
<td>FIELDS</td>
<td>SYSIBM.SYSFIELDS</td>
</tr>
<tr>
<td>FOREIGN</td>
<td>SYSIBM.SYSFOREIGNKEYS</td>
</tr>
<tr>
<td>INDEXES</td>
<td>SYSIBM.SYSINDEXES</td>
</tr>
<tr>
<td>INDEXH</td>
<td>SYSIBM.SYSINDEXES_HIST</td>
</tr>
<tr>
<td>INDEXPA</td>
<td>SYSIBM.SYSINDEXPART</td>
</tr>
<tr>
<td>INDEXPH</td>
<td>SYSIBM.SYSINDEXPART_HIST</td>
</tr>
<tr>
<td>INDEXSH</td>
<td>SYSIBM.SYSINDEXSTATS_HIST</td>
</tr>
<tr>
<td>INDEXST</td>
<td>SYSIBM.SYSINDEXSTATS</td>
</tr>
<tr>
<td>IPLIST</td>
<td>SYSIBM.IPLIST</td>
</tr>
<tr>
<td>IPNAMES</td>
<td>SYSIBM.SYSIPNAMES</td>
</tr>
<tr>
<td>JARCONT</td>
<td>SYSIBM.SYSJARCONTENTS</td>
</tr>
<tr>
<td>JAROBJT</td>
<td>SYSIBM.SYSJAROBJECTS</td>
</tr>
<tr>
<td>JAVAPTH</td>
<td>SYSIBM.SYSVARIABLEAUTH</td>
</tr>
<tr>
<td>JAVOPTS</td>
<td>SYSIBM.SYSJAVAOPTS</td>
</tr>
<tr>
<td>KCOLUSE</td>
<td>SYSIBM.SYSKEYCOLUSE</td>
</tr>
<tr>
<td>KEYS</td>
<td>SYSIBM.SYSKEYS</td>
</tr>
<tr>
<td>KEYTDST</td>
<td>SYSIBM.SYSKEYTGTDISTSTATS</td>
</tr>
<tr>
<td>KEYTGD</td>
<td>SYSIBM.SYSKEYTGTDIST</td>
</tr>
<tr>
<td>KEYTGDH</td>
<td>SYSIBM.SYSKEYTGTDIST_HIST</td>
</tr>
<tr>
<td>KEYTGHS</td>
<td>SYSIBM.SYSKEYTARGETS_HIST</td>
</tr>
<tr>
<td>KEYTGST</td>
<td>SYSIBM.SYSKEYTARGETSTATS</td>
</tr>
<tr>
<td>KEYTGT</td>
<td>SYSIBM.SYSKEYTARGETS</td>
</tr>
<tr>
<td>LOBSTAH</td>
<td>SYSIBM.SYSLOBSTATS_HIST</td>
</tr>
<tr>
<td>LOBSTAT</td>
<td>SYSIBM.SYSLOBSTATS</td>
</tr>
<tr>
<td>Option</td>
<td>Catalog table</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>LOCATIO</td>
<td>SYSIBM.LOCATIONS</td>
</tr>
<tr>
<td>LULIST</td>
<td>SYSIBM.LULIST</td>
</tr>
<tr>
<td>LUMODES</td>
<td>SYSIBM.LUMODES</td>
</tr>
<tr>
<td>LUNAMES</td>
<td>SYSIBM.LUNAMES</td>
</tr>
<tr>
<td>MODESEL</td>
<td>SYSIBM.MODESELECT</td>
</tr>
<tr>
<td>OBDS</td>
<td>SYSIBM.SYSOBDS</td>
</tr>
<tr>
<td>OBJROL</td>
<td>SYSIBM.SYSOBJROLEDEP</td>
</tr>
<tr>
<td>PACKAGE</td>
<td>SYSIBM.SYSPACKAGE</td>
</tr>
<tr>
<td>PACKAUT</td>
<td>SYSIBM.SYSPACKAUTH</td>
</tr>
<tr>
<td>PACKCPY</td>
<td>SYSIBM.SYSPACKCOPY</td>
</tr>
<tr>
<td>PACKDEP</td>
<td>SYSIBM.SYSPACKDEP</td>
</tr>
<tr>
<td>PACKLIS</td>
<td>SYSIBM.SYSPACKLIST</td>
</tr>
<tr>
<td>PACKSTM</td>
<td>SYSIBM.SYSPACKSTMT</td>
</tr>
<tr>
<td>PARMS</td>
<td>SYSIBM.SYSPARMS</td>
</tr>
<tr>
<td>PENDDDL</td>
<td>SYSIBM.SYSPENDINGDDL</td>
</tr>
<tr>
<td>PKSYSTE</td>
<td>SYSIBM.SYSPKSYSTEM</td>
</tr>
<tr>
<td>PLAN</td>
<td>SYSIBM.SYSPLAN</td>
</tr>
<tr>
<td>PLANAUT</td>
<td>SYSIBM.SYSPLANAUTH</td>
</tr>
<tr>
<td>PLANDEP</td>
<td>SYSIBM.SYSPLANDEP</td>
</tr>
<tr>
<td>PLSYTE</td>
<td>SYSIBM.SYSPLSYSTEM</td>
</tr>
<tr>
<td>PROCEDU</td>
<td>SYSIBM.SYSPROCEDURES</td>
</tr>
<tr>
<td>QUEROPT</td>
<td>SYSIBM.SYSQUERYOPTS</td>
</tr>
<tr>
<td>QUERPLN</td>
<td>SYSIBM.SYSQUERYPLAN</td>
</tr>
<tr>
<td>QUERY</td>
<td>SYSIBM.SYSQUERY</td>
</tr>
<tr>
<td>RELS</td>
<td>SYSIBM.SYSRELS</td>
</tr>
<tr>
<td>RESAUTH</td>
<td>SYSIBM.SYSRESAUTH</td>
</tr>
<tr>
<td>ROLES</td>
<td>SYSIBM.SYSROLES</td>
</tr>
<tr>
<td>ROUTINA</td>
<td>SYSIBM.SYSROUTINEAUTH</td>
</tr>
<tr>
<td>ROUTINE</td>
<td>SYSIBM.SYSROUTINES</td>
</tr>
<tr>
<td>ROUTOPT</td>
<td>SYSIBM.SYSROUTINES_OPTS</td>
</tr>
<tr>
<td>ROUTSRC</td>
<td>SYSIBM.SYSROUTINES_SRC</td>
</tr>
<tr>
<td>Option</td>
<td>Catalog table</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>SCHEMAA</td>
<td>SYSIBM.SYSSCHEMAAUTH</td>
</tr>
<tr>
<td>SEQAUTH</td>
<td>SYSIBM.SYSSEQUENCEAUTH</td>
</tr>
<tr>
<td>SEQDEP</td>
<td>SYSIBM.SYSSEQUENCESDEP</td>
</tr>
<tr>
<td>SEQUENC</td>
<td>SYSIBM.SYSSEQUENCES</td>
</tr>
<tr>
<td>STMT</td>
<td>SYSIBM.SYSSTMT</td>
</tr>
<tr>
<td>STOGROU</td>
<td>SYSIBM.SYSSTOGROUP</td>
</tr>
<tr>
<td>STRINGS</td>
<td>SYSIBM.SYSSTRINGS</td>
</tr>
<tr>
<td>SYNONYM</td>
<td>SYSIBM.SYSSYNONYMS</td>
</tr>
<tr>
<td>TABAUTH</td>
<td>SYSIBM.SYSTABAUTH</td>
</tr>
<tr>
<td>TABCNST</td>
<td>SYSIBM.SYSTABCONST</td>
</tr>
<tr>
<td>TABLEPA</td>
<td>SYSIBM.SYSTABLEPART</td>
</tr>
<tr>
<td>TABLES</td>
<td>SYSIBM.SYSTABLES</td>
</tr>
<tr>
<td>TABLESH</td>
<td>SYSIBM.SYSTABLES_HIST</td>
</tr>
<tr>
<td>TABLESP</td>
<td>SYSIBM.SYSTABLESPACE</td>
</tr>
<tr>
<td>TABPRTH</td>
<td>SYSIBM.SYSTABLEPART_HIST</td>
</tr>
<tr>
<td>TABSTAH</td>
<td>SYSIBM.SYSTABSTATS_HIST</td>
</tr>
<tr>
<td>TABSTAT</td>
<td>SYSIBM.SYSTABSTATS</td>
</tr>
<tr>
<td>TBLPROF</td>
<td>SYSIBM.SYSTABLES_PROFILES</td>
</tr>
<tr>
<td>TRIGGER</td>
<td>SYSIBM.SYSTRIGGERS</td>
</tr>
<tr>
<td>USERAUT</td>
<td>SYSIBM.SYSUSERAUTH</td>
</tr>
<tr>
<td>USERNAM</td>
<td>SYSIBM.USERNAMES</td>
</tr>
<tr>
<td>VIEWDEP</td>
<td>SYSIBM.SYSVIEWDEP</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.SYXMLRELS</td>
</tr>
<tr>
<td>XMLSTR</td>
<td>SYSIBM.SYXMLSTRINGS</td>
</tr>
<tr>
<td>XSROBJ</td>
<td>SYSIBM.XSROBJECTS</td>
</tr>
</tbody>
</table>
JCL Generation product options

A keyword in the ALTER, CATALOG MANAGER, CHANGE MANAGER, and DASD MANAGER PLUS installation options modules, POFDS, specifies the product options file (POF).

The POF file is an 80-character sequential file that contains keywords and values for the JCL Generation options. The file is located in the HLQ.UBMCCNTL 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.

Example of product options

This section provides an example of a product options file.

* Figure 177: Product options file

```plaintext
*-------------------------------------------------------------
*  POF WRITTEN FROM VERSION:  V11.02.00
*  FORMAT:
*  # KEYWORD=PARAM  COLUMNS 1-80.
*  # PARAM SYNTAX:
*    # VALUE - EVERYTHING AFTER THE = IS CONSIDERED THE VALUE.
*    # LEADING AND TRAILING BLANCS 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 PARAM
*    # ON NEXT LINE.
*    # BLANK LINES ARE IGNORED.
*    # ASTERISK IN COLUMN 1 INDICATES THAT LINE IS A COMMENT.
*-------------------------------------------------------------
POFDATE = 2014/05/16 12:14:10
ACM_AMS = Y
ACM_ANALYSIS_SYSOUT = A
ACM_BASDIAG = SYSOUT
ACM_BRPTDIAG = SYSOUT
ACM_BRPTDSN = '&PREFIX..BASELINE.REPORT'
```
Example of product options

ACM_CDLDSDN = '&PREFIX..&SSID..CDL(CDL)
ACM_CDLP = 15
ACM_CDLS = 5
ACM_CDLY = SYSDA
ACM_CMPDIAG = SYSOUT
ACM_CPLCDLO = '&PREFIX..&SSID..CDL(CDL)
ACM_CPLDIAG = SYSOUT
ACM_CPLWDSN = '&PREFIX..&SSID..&TASKID
ACM_CPLWDSN0 = '&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_JDSNCPLC = '&PREFIX..EXEC(&WORKID)
ACM_JDSNCPL = '&PREFIX..EXEC(&WORKID)
ACM_JDSN = '&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_WLORDER = N
ACM_WLORDERMSG = Y
ACM_WLPS = 15
ACM_WLSS = 5
ACM_WLU = SYSDA
ADDLOAD1 = BMCRMD.V810INST.NONSMPE_LOAD,(R)
ADDLOAD2 = BMCRMD.V810INST.UBMCLINK,(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=

416  CATALOG MANAGER for DB2 User Guide
Example of product options
Example of product options

Appendix F  JCL Generation product options 419
Example of product options

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
PCPY2_STORCLASS =
PCPY2_STORCLASS_ALT =
PCPY2_SUPPRESS_SUFF = N
PCPY2_THRESH = 0
PCPY2_UNIT = SYSDA
PCPY2_UNIT_ALT =
PRE_JOBSTEP_INCLUDE =
PROC_BMCCHECK_NAME =
PROC_BMCCHECK_STEP =
PROC_BMCCOPY_NAME =
PROC_BMCCOPY_STEP =
PROC_BMCCPRS_NAME =
PROC_BMCCPRS_STEP =
PROC_BMCLOAD_NAME =
PROC_BMCLOAD_STEP =
PROC_BMCRECOVER_NAME =
PROC_BMCRECOVER_STEP =
PROC_BMCREORG_NAME =
PROC_BMCREORG_STEP =
PROC_BMCSSTATS_NAME =
PROC_BMCSSTATS_STEP =
PROC_BMCSTOP_NAME =
PROC_BMCSTOP_STEP =
PROC_BMCTRIG_NAME =
PROC_BMCTRIG_STEP =
PROC_BMCUNLOAD_NAME =
PROC_BMCUNLOAD_STEP =
PROC_BMCUPRS_NAME =
PROC_BMCUPRS_STEP =
PROC_DSNUTILB_NAME =
PROC_DSNUTILB_STEP =
PROC_DSN1COPY_NAME =
PROC_DSN1COPY_STEP =
PROC_GEN_SET_VAR = N
PROC_IDCAMS_NAME =
PROC_IDCAMS_STEP =
PROC_IEFBR14_NAME =
PROC_IEFBR14_STEP =
PROC_TSO_NAME =
PROC_TSO_STEP =
PROC_USE = N
PROC_USER_DEF_STEP =
PROC_USER_DEFINED =
PUNCH_DATACLASS =
PUNCH_EXPDT =
Example of product options

```plaintext
PUNCH_MGMTCLASS =
PUNCH_PREFIX = &PREFIX..&WKID..&STEPN
PUNCH_PRIQTY = 1
PUNCH_RETPD =
PUNCH_SECQTY = 1
PUNCH_STORCLASS =
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_MAPDB =
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
SQLEXP_LOAD = &PREFIX.LOAD,(R)
SRTOUT_DATACLASS =
SRTOUT_DATACLASS_ALT =
```

Appendix F  JCL Generation product options 421
Example of product options
Descriptions of product option keywords

This section provides descriptions of the keywords in the product options file.

See also “Example of product options” on page 415.

Note
The , (R) in the variable syntax indicates that the specified value will refresh the existing value of the variable in the user’s ISPF profile data set when the POFDATE is later than the previous POFDATE stored in the user’s ISPF profile.
2MEGSQL=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_ANALYSIS_SYSOUT =A

For ALTER and CHANGE MANAGER, this keyword sets the default output class to a value other than X.

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Generates AMS statements (IDCAMS DELETE and DEFINE) in a worklist</td>
</tr>
<tr>
<td>N</td>
<td>Generates a -STOP worklist command that enables you to complete the DELETE and DEFINE commands before the DB2 object CREATE commands that are located later in the worklist</td>
</tr>
</tbody>
</table>

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.

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_JDSNBJBG=’&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_JDSNCPOLO='&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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Executes 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.</td>
</tr>
<tr>
<td>N</td>
<td>Executes the worklist sequentially, even if the required parallelism worklist commands are included in the worklist</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Writes tracing records to the AEXPTRAC output data set AEXPTRAC is dynamically allocated and the output is written to SYSOUT.</td>
</tr>
<tr>
<td>N</td>
<td>Does not use tracing, even if an //AEXPTRAC DD statement is specified in the JCL</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Sorts the worklist by each table’s cardinality in descending sequence</td>
</tr>
<tr>
<td>N</td>
<td>Sorts the worklist by table order in ascending sequence, according to the table owner and table name</td>
</tr>
<tr>
<td>A</td>
<td>Sorts the worklist by either table cardinality or by table order, depending on whether the worklist is processed in parallel</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Blank</td>
<td>Generates the objects in the worklist in an unsorted, random order</td>
</tr>
</tbody>
</table>

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= HLQ.UBMCLINK

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 formats are yyddd 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 9999.

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_DATAC=

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

ASU_XP_LOGD_MGMTC=

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_SECQTY=2

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

ASU_XP_LOGD_STORC=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 formats 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..&OBNO

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 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=ACK$OPTS**

This keyword specifies the name of theCATALOG MANAGER 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.

---

**Tip**

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

**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.

---

**Tip**

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

---

**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.

---

**Tip**

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

This keyword specifies the name of the LINK library for the CATALOG MANAGER 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 CATALOG MANAGER 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 formats 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-IEFBR14 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 formats 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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Does not perform data set sizing</td>
</tr>
<tr>
<td>C</td>
<td>Uses IBM RUNSTATS to perform data set sizing by using statistics from the DB2 catalog</td>
</tr>
<tr>
<td>B</td>
<td>Uses BMCSTATS to perform data set sizing by using the statistics from the BMC DASD MANAGER PLUS product tables</td>
</tr>
<tr>
<td>O</td>
<td>Physically and randomly samples the VSAM objects to estimate data set sizes</td>
</tr>
</tbody>
</table>
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.

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

DB2LOAD

This keyword specifies the name of the DB2 LOAD library.

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

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.

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 formats are yyyy/ddd 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..&OBNOD

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.
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>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.</td>
</tr>
<tr>
<td>N</td>
<td>Indicates to display the object name in a panel.</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Indicates not to display an autotab character.</td>
</tr>
<tr>
<td>any character other than N</td>
<td>Displays as the autotab character.</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Replaces characters at the left end (beginning) of the name.</td>
</tr>
<tr>
<td>M</td>
<td>Replaces characters in the middle of the name.</td>
</tr>
<tr>
<td>E</td>
<td>Replaces characters at the right end (end) of the name.</td>
</tr>
</tbody>
</table>

**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_DBG=\texttt{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=\texttt{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).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{N}</td>
<td>Does not allow an object to be dropped, if no image copies of the object exist</td>
</tr>
<tr>
<td>\texttt{Y}</td>
<td>Allows an object to be dropped, even if no image copies of the object exist</td>
</tr>
</tbody>
</table>

DSNCHECK44=\texttt{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 \texttt{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 \texttt{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 formats 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 formats are yyddd or yyyy/ddd.

FCPY_MGMTCLASS

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..&\OBNOD..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.

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 99999.

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.

Note
If GDG_MODEL= NONE, DCB= model.dataSetName is omitted from the JCL for the 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.

HASHWARNRC

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.

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.
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**

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

**LL_LINK**

These keywords specify the LLQ for the LOAD library data sets for the batch ISPF environment.

**LL_MLIB**

These keywords specify the LLQ for the message data sets for the batch ISPF environment.

**LL_PLIB**

These keywords specify the LLQ for the panel and Help library data sets for the batch ISPF environment.
These keywords specify the LLQ for the ISPF skeleton data sets for the batch ISPF environment.

These keywords specify the LLQ for the ISPF table data sets for the batch ISPF environment.

These keywords specify the LLQ for the utility generation data sets for the batch ISPF environment.

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 sets this value to BMC. If you chose not to use the feature, the Installation System sets 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.

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.

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..&OBNO\_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..&OBNOD..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 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 CHECK PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

PROC_BMCCCOPY_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_BMCCCOPY_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 BMCSTATS component of 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 BMCSTATS component of 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_BMCCTRL_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 BMCTRLG component of the BMC DASD MANAGER PLUS for DB2 utility.

This keyword is not included in the AJXPOFIN input stream.

**PROC_BMCCTRL_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 BMCTRLG 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_BMCURS_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 9999.

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_MAPDB**

This keyword specifies the mapping database that ALTER and CHANGE MANAGER provide to the IBM REORG utility as an override to the value of the DB2 subsystem parameter REORG_MAPPING_DATABASE. CATALOG MANAGER and DASD MANAGER PLUS do not use this value. The name can be from 1 to 8 characters long, and cannot include symbolic variables.
**Note**

REORG_MAPDB applies to the dynamically and non-dynamically allocated data sets that the IBM REORG utility uses.

REORG_MAPDB also applies to the dynamically allocated data sets that the BMC REORG PLUS utility uses.

**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.

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 (0) 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. Zero (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).

SYNDELETE=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.

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Does not use a 7-track drive</td>
</tr>
<tr>
<td>C</td>
<td>Uses odd parity, conversion on, and translation off</td>
</tr>
<tr>
<td>E</td>
<td>Uses even parity, conversion off, and translation off</td>
</tr>
<tr>
<td>T</td>
<td>Uses odd parity, conversion off, and translation on</td>
</tr>
<tr>
<td>ET</td>
<td>Uses even parity, conversion off, and translation on</td>
</tr>
<tr>
<td>COMP</td>
<td>Uses data compression on</td>
</tr>
<tr>
<td>NOCOMP</td>
<td>Uses data compression off</td>
</tr>
</tbody>
</table>

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 9999.

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_HLQ**

This keyword specifies the high-level qualifier (HLQ) used by user defined data sets. This HLQ is also used for the runtime enablement (RTE) data sets.

**USER_VARS**

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

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:

- Line identifier, command identifier, command sequence number, and short command text on the first line
- Blank in the line identifier column, with long command text on continuation lines
- 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 (Execute a BMC utility)

The -BMCU command invokes a BMC utility to run as a subtask.

Figure 178: -BMCU command

-BMCU 000004 ASUSMAIN
BMCSTATS INDEX QZU.QZUX01_D30S05T01
EVENTS N

Note
The utility command starts on the second line of the statement. The first line is reserved for parameters passed to the utility, such as NEW, TERM, MAINT, NEW/RESET, and TERM/RESET.
The text portion of the statement identifies the invoked program name and the parameters passed to the program.
DASD MANAGER PLUS generates this command.

-DBG (Debug)

The -DBG command turns on or off the DEBUG and FLOW options.

The following keywords are valid:

- DEBUG
- DEBUGOFF
- FLOW
- FLOWOFF

Note
The Administrative products do not generate this command. If necessary, contact BMC Customer Support, who will direct you to add this command to your worklist to resolve an issue.
**-DSN1 (IBM DSN1COPY utility)**

The -DSN1 command invokes the IBM DSN1COPY utility.

If you need to code this command yourself (see Figure 179 on page 496), 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 DSNAME 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 storage group defined, and the Execution component looks up the high-level qualifier. (Parm='parameterString').
- Do not define a SYSUT1 DD in the JCL.

**Figure 179: -DSN1 command**

```
-DSN1 000004
DSN1DDIN DS1I0001
DEAEACAT.DSNDBD.WZ2130.WZS12130.I0001.A237
DSN1DDOU DS1O0001
PARM='PAGESIZE(4K)'
```

For more information, see the documentation for the IBM utilities.

**Note**

CATALOG MANAGER and DASD MANAGER PLUS generate this command.

---

**-MERG (IBM MERGECOPY utility)**

The -MERG command invokes the IBM MERGECOPY utility.

This command and its parameters are passed to DB2.

**Figure 180: -MERG command**

```
MERG 050000
MERCOPY 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 (IBM MODIFY utility)

The -MODI command invokes the IBM MODIFY RECOVERY or MODIFY STATISTICS utility.

**Figure 181 on page 497** shows an example of the -MODI command for the MODIFY RECOVERY utility.

**Figure 181: -MODI command—MODIFY RECOVERY**

```
-MODI 100001
MODIFY RECOVERY
  TABLESPACE ACTQX18.ACTS0118
  DELETE AGE(*)
```

**Figure 182 on page 497** shows an example of the -MODI command for the MODIFY STATISTICS utility.

**Figure 182: -MODI command—MODIFY STATISTICS**

```
-MODI 000002
MODIFY STATISTICS
  TABLESPACE ACTQX18.ACTS0118
  DELETE ALL AGE(*)
```

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 183 on page 497** shows an example of the -NOOP command.

**Figure 183: -NOOP command**

```
-NOOP 000010
  SELECT
```
LENGTH( DBNAME ) AS DBLENG
  .
  .  DBNAME
  .
  .  LENGTH( NAME) AS TSLENG
  .  NAME
  .
  .  PARTITIONS
  .  CURRENT SERVER AS CURRSERV
  .
  .  TS.
  FROM SYIBM.SYSTABLESPACE TS
  WHERE DBNAME = 'ACTQX11' AND NAME = 'ACTS0311'
  WITH UR

Note
CHANGE MANAGER generates this command.

-QUI (IBM QUIESCE utility)

The -QUI command invokes the IBM DB2 QUIESCE utility.

Following is an example of the -QUI command.

**Figure 184: -QUI command**

```
-QUI 001020
QUIESCE TABLESPACE ASUDBU04.U041
    TABLESPACE ASUDBU04.U042
    TABLESPACE ASUDBU04.U051
    TABLESPACE ASUDBX1.WS11
    TABLESPACE ASUDBX1.WS32
```

For more information, see the documentation for the IBM utilities.

Note
CATALOG MANAGER and DASD MANAGER PLUS generate this command.

-REPO (IBM REPORT utility)

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 185 on page 498** shows an example of the -REPO command.

**Figure 185: -REPO command**

```
-REPO 000001
REPORT RECOVERY TABLESPACE
    ACTQX16.ACTS0116
    CURRENT
    SUMMARY
```
For more information, see the documentation for the IBM utilities.

**Note**
CHANCE MANAGER and DASD MANAGER PLUS generate this command.

**-REPX (IBM REPAIR utility)**

The -REPX command invokes the IBM 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 groups multiple ALTER statements (for table space partitions and their indexes) within a single -SQL command to reduce the total number of steps in the worklist and increase worklist performance.

**Figure 186: -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 (IBM STOSPACE utility)**

The -STOS command invokes the IBM STOSPACE utility.
**Figure 187: -STOS command**

```
-STOS 010000

STOSPACE STOGROUP DEV070
```

For more information, see the documentation for the IBM utilities.

---

**Note**

CHANGE MANAGER and DASD MANAGER PLUS generate this command.
Commands

This section describes the various types of commands that you can access from CATALOG MANAGER.

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 65 on page 502 lists the commands.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| ACCEL   | From a table list, adds a table to an accelerator. From an accelerator list (ACC), enables you to control an accelerator. From an accelerator table list (ACCTB), provides access to DB2 accelerator commands that enable you to:  
- Modify distribution or organizing keys  
- Move data from DB2 to an accelerator  
- Return the current definition and status information of an accelerator  
- Copy data from DB2 to an accelerator  
- Remove data and a table from an accelerator  
- Restore data that was moved to an accelerator by an ARCHIVE TABLES to DB2  
- Enable or disable use of a loaded table on the accelerator  
- Enable or disable incremental updates for a table |
<p>| 2WL     | When used from the SQL_Table list, creates a worklist from the selected SQL member |
| ANALYZE | Displays a SELECT, INSERT, UPDATE, DELETE, or DECLARE CURSOR SQL statement from the CATALOG MANAGER SQL_Table. You can invoke the BMC SQL Explorer Analysis function to analyze the SQL. |
| APPLY   | Creates SQL statements by using a model with host variables into which CATALOG MANAGER will substitute values from the DB2 catalog. The name of the host variable should correspond to the column name of the value to be substituted. |
| AUDIT   | Displays rows from the CATALOG MANAGER Audit Log table |
| BATCH   | Generates JCL for a CATALOG MANAGER job to run in batch mode. Use BATCH when lengthy processing time makes running the job online undesirable. |
| BROWSE  | Invokes the CATALOG MANAGER data browsing feature to browse data in the selected table or view |
| CANCEL  | Terminates the current display without saving any changes that might have been made |
| CASCADE | Displays the privileges that the REVOKE command would revoke for the specified user |</p>
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOGHELP</td>
<td>Displays information about columns in the DB2 catalog tables</td>
</tr>
</tbody>
</table>
| CLIPBOARD   | Displays the Confirm SQL panel for the contents of the CATALOG MANAGER clipboard  
On this panel, you can edit, save, and execute the SQL from the clipboard.  
CATALOG MANAGER saves changes in the CATALOG MANAGER SQL_Table, not in the clipboard. |
| CLIST       | Allows CATALOG MANAGER to execute a CLIST that is available but is not in the commands table                                                                                                             |
| CMD         | See COMMANDS                                                                                                                                                                                               |
| COMMANDS    | Lists the valid CATALOG MANAGER commands for the current list list type  
For example, different commands are listed for table spaces and columns.                                                                                                    |
| COPYAUTHS   | Generates SQL that will copy authorizations from a source user to a target user  
and will grant authorizations from a source object to a target object                                                                                             |
| CUSTOMIZE   | Allows you to customize the CATALOG MANAGER Primary Menu panel to include a subset of the options                                                                                                         |
| D           | Displays the catalog row for the selected object                                                                                                                                                    |
| DCL         | Generates GRANTs for explicit privileges that are held on an object or by a user                                                                                                                           |
| DDL         | Displays the DDL for the selected object                                                                                                                                                                |
| DES         | Displays a subset of the description that is provided by the DESCRIBE command for tables, databases, and DBRM packages                                                                                     |
| DESCRIBE    | Displays catalog information about the selected object  
In some instances, the information might come from more than one catalog table;  
for example, a DESCRIBE command on a database will show the plans that are associated with the database.                                    |
| DOPTS       | Displays the installation options for this session of CATALOG MANAGER                                                                                                                                   |
| DROPRECOVERY| Displays a list of objects that CATALOG MANAGER dropped with Recovery On specified, and allows you to select an object to recover                                                                        |
| EDIT        | Invokes the CATALOG MANAGER data editing feature to edit data in the selected table or view                                                                                                               |
| ENVIRONMENT | Displays some of the CATALOG MANAGER variables  
You can use the ENVIRONMENT command to verify the plans and modules that are in use.                                                                                             |
<p>| EXPLAIN     | Displays rows from the ownerName.PLAN_TABLE for the object                                                                                                                                               |
| F FIND       | Find strings in customizable list columns                                                                                                                                                                |</p>
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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. 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 (HDDL) for the selected object. CATALOG MANAGER options determine which dependent objects are included in the dependent DDL.</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>IMPORT</td>
<td>Copies SQL from a PDS into the CATALOG MANAGER SQL_TABLE. 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. 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. LOGRBA also records this information in the CATALOG MANAGER Drop Recovery Log.</td>
</tr>
<tr>
<td>MAINTAIN</td>
<td>Displays the Log Maintenance Menu, on which you can browse or purge the Session Log, DDL Audit Log, and Drop Recovery Log.</td>
</tr>
<tr>
<td>MDDL</td>
<td>Generates DDL for all objects that you have listed in the command into 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>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>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PRO PROFILE PROFILES</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>
<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>TSO POFRESET POF(dataSetName(POFMember)) resets all of the ISPF variables in the ISPF profile with the variables in the specified initial or user POF name. The POFRESET CLIST enables these commands. The CLIST is located in the HLQ.BMCCLLIB library.</td>
</tr>
<tr>
<td>X</td>
<td>When entered in the Cmd column of a list line, excludes objects from being processed by the command that you specify on the Command 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 Cmd column of a list line from being processed by the command that you specify on the Command line.</td>
</tr>
<tr>
<td>X nnn</td>
<td>Excludes the object in a list line and the objects in the following nnn list lines from being processed by the command that you specify on the Command line.</td>
</tr>
</tbody>
</table>
DB2 action commands

CATALOG MANAGER generates the following DB2 commands.

To execute a command, type the command in the **Cmd** column next to an appropriate item on a list. Table 66 on page 506 lists the DB2 action commands.

**Table 66: DB2 action commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| ACTIVATE | For a native SQL procedure, generates an SQL ALTER PROCEDURE ACTIVATE statement  
ACTIVATE is valid only for stored procedures in which ORIGIN is N. |
| ALTER | Generates an SQL ALTER statement for the object |
| BIND | Displays the Bind 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 bind plans, packages, or DBRMs. |
| BINDCOPY | Binds all packages in a collection ID (CI) into a different CI, or binds all packages in a list into a single CI |
| BINDDEPLOY | Displays the Bind Deploy Package panel, from which you can specify the locations to which you want to generate BIND commands  
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.  
For a native SQL procedure, the command generates a DSN BIND PACKAGE DEPLOY statement.  
BINDDEPLOY is valid only for packages in which TYPE is N. |
| COMMENT | Displays a panel to generate SQL to create or replace a comment on an alias, table, distinct type, function, index, procedure, trigger, or view |
| CONNECT | Enables you to remain in your current CATALOG MANAGER session and access another DB2 subsystem on the same or another z/OS system |
| COUNT | Executes an SQL COUNT(*) statement for an object and displays the number of rows in the object |
| CREATE | Displays a panel from which you can input options to create an object  
If entered on a list line, CREATE works as a CREATE LIKE of the object. |
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2COMMAND</td>
<td>Lists the DB2 commands that have been saved. Enter DB2COMMAND PROMPT 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 DISPLAY</td>
<td>Displays the status of the selected object.</td>
</tr>
<tr>
<td>DIS DATABASE</td>
<td>Displays the status of the selected database.</td>
</tr>
<tr>
<td>DISTHREAD</td>
<td>Displays active threads for the current subsystem.</td>
</tr>
<tr>
<td>DROP</td>
<td>Generates an SQL DROP statement for the specified object and displays the Confirm SQL panel. From this panel, you can select actions and options, including the drop recovery option. When you issue the DROP command on the Pending DDL List panel, CATALOG MANAGER generates the ALTER TABLESPACE DROP PENDING CHANGES statement.</td>
</tr>
<tr>
<td>DSNZPARM</td>
<td>Executes the IBM DSNWZP stored procedure and formats the information for display.</td>
</tr>
<tr>
<td>EXCHANGE</td>
<td>Exchanges data between two tables, one of which must be a clone of the other table.</td>
</tr>
<tr>
<td>FREE</td>
<td>Generates SQL to free active versions of plans or packages.</td>
</tr>
<tr>
<td>FREE ALL ALL</td>
<td>Generates SQL to free all plans or packages in a list and all versions of the plans or packages. FREE ALL ALL specifies the DB2 PLANMGMTCSCOPE(ALL) clause.</td>
</tr>
<tr>
<td>FREE INACT</td>
<td>Generates SQL to free inactive versions of plans or packages in a list. FREE INACT or FREE INACTIVE specifies the DB2 PLANMGMTCSCOPE(INACTIVE) clause.</td>
</tr>
<tr>
<td>GRANT</td>
<td>Generates SQL to grant privileges.</td>
</tr>
<tr>
<td>INSERT</td>
<td>Builds an SQL INSERT statement template for a table or view, and invokes the ISPF editor to allow you to customize the statement.</td>
</tr>
<tr>
<td>LABEL</td>
<td>Generates SQL to grant privileges.</td>
</tr>
<tr>
<td>QCONNECT</td>
<td>Displays the Connections List panel, which lists saved connections from the current SSID. If no saved connections exist, QCONNECT displays the Change Access panel, from which you can type parameters for a connection.</td>
</tr>
</tbody>
</table>
### 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 67 on page 509** lists the utility commands.
### Table 67: Utility commands

<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 CHECK PLUS jobs with the CHECK DATA command</td>
</tr>
<tr>
<td>BMCCHECK INDEX</td>
<td>Displays panels from which you can input options and generate 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 CHECK PLUS jobs with the CHECK INDEX command</td>
</tr>
<tr>
<td>BMCC HTS</td>
<td>Displays panels from which you can input options and generate CHECK PLUS jobs with the CHECK TABLESPACE command</td>
</tr>
<tr>
<td>BMCCOPY</td>
<td>Displays panels from which you can input options and generate COPY PLUS jobs</td>
</tr>
<tr>
<td>BMCCOPY INDEX</td>
<td>Displays panels from which you can input options and generate COPY PLUS jobs</td>
</tr>
<tr>
<td>BMCCOPY IX</td>
<td>Displays panels from which you can input options and generate COPY PLUS jobs</td>
</tr>
<tr>
<td>BMCEXPLOR E</td>
<td>Calls the SQL Explorer interface</td>
</tr>
<tr>
<td>BMCLOAD</td>
<td>Displays panels from which you can input options and generate LOADPLUS jobs</td>
</tr>
<tr>
<td>BMCREBUILD</td>
<td>Displays panels from which you can input options and generate RECOVER PLUS jobs with the REBUILD INDEX command</td>
</tr>
<tr>
<td>BMCREBUILD INDEX</td>
<td>Displays panels from which you can input options and generate RECOVER PLUS jobs with the REBUILD INDEX command</td>
</tr>
<tr>
<td>BMCREBUILD IX</td>
<td>Displays panels from which you can input options and generate RECOVER PLUS jobs with the REBUILD INDEX command</td>
</tr>
<tr>
<td>BMCRECOVER</td>
<td>Displays panels from which you can input options and generate RECOVER PLUS jobs</td>
</tr>
<tr>
<td>BMCRECOVER INDEX</td>
<td>Displays panels from which you can input options and generate RECOVER PLUS jobs</td>
</tr>
<tr>
<td>BMCRECOVERY IX</td>
<td>Displays panels from which you can input options and generate REORG PLUS jobs</td>
</tr>
<tr>
<td>BMCREORG</td>
<td>Displays panels from which you can input options and generate REORG PLUS jobs</td>
</tr>
<tr>
<td>BMCREORG INDEX</td>
<td>Displays panels from which you can input options and generate REORG PLUS jobs</td>
</tr>
<tr>
<td>BMCREORG IX</td>
<td>Displays panels from which you can input options and generate REORG PLUS jobs</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>Displays panels from which you can input options and generate BMCSTATS jobs</td>
</tr>
<tr>
<td>BMCSTATS IX</td>
<td>Displays panels from which you can input options and generate BMCSTATS jobs</td>
</tr>
<tr>
<td>BMCSTATS ix</td>
<td>BMCSTATS is a component of the DASD MANAGER PLUS product.</td>
</tr>
<tr>
<td>BMCUHIST</td>
<td>Displays information from the BMC Utility History table</td>
</tr>
<tr>
<td>BMCUHIST IX</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 UNLOAD PLUS jobs</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</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. From this list, you can terminate a utility. To limit the amount of information displayed by the command, include the <code>userID.utilityID</code> 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. To limit the amount of information displayed by the command, include the <code>userID.utilityID</code> 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 DASD MANAGER PLUS product to report exceptions on the object.</td>
</tr>
<tr>
<td>EXEC</td>
<td>Displays panels from which you can input options for the IBM EXEC SQL utility.</td>
</tr>
<tr>
<td>LOAD</td>
<td>Displays panels from which you can input options for the IBM LOAD utility.</td>
</tr>
<tr>
<td>MERGECOPY</td>
<td>Displays panels from which you can input options for the IBM MERGECOPY utility.</td>
</tr>
<tr>
<td>MODIFY</td>
<td>Displays panels from which you can input options for the IBM MODIFY utility.</td>
</tr>
<tr>
<td>MODIFYRECOVERY</td>
<td>Displays panels from which you can input options for the IBM MODIFY utility with the RECOVERY option.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</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>REBUILD IX</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>RECOVER IX</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>REORG IX</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>REPORT IX</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 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. To limit the amount of information displayed by the command, include the user1D.utility1D 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. The command is valid only from a utility list.</td>
</tr>
<tr>
<td>UNLOAD</td>
<td>Displays panels from which you can input options for the IBM UNLOAD utility</td>
</tr>
</tbody>
</table>
Utility list commands

The following table 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 “Utility commands” on page 508) on a list.

Table 68: 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 Object names that have been combined are no longer displayed on the utility statement list.</td>
</tr>
<tr>
<td>?</td>
<td>Displays a list of available commands for the Cmd column 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 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 After you select a utility profile on the Utility List panel, you cannot reset the profile until you exit the panel or issue the RESET command on the Command line.</td>
</tr>
</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 69 on page 513** lists the statistics commands.

**Table 69: Statistics commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATSTATS</td>
<td>Executes the SQL COUNT (*) statement for objects in the DB2 catalog</td>
</tr>
<tr>
<td>COLSTATS</td>
<td>Displays statistics from the SYSIBM.SYSCOLSTATS catalog table for the selected table or column</td>
</tr>
<tr>
<td>DEST</td>
<td>Displays statistical information and catalog row information for the selected object</td>
</tr>
<tr>
<td>DESTATISTICS</td>
<td></td>
</tr>
<tr>
<td>DISTSTATS</td>
<td>Displays statistics from the SYSIBM.SYSCOLDIST or SYSIBM.SYSCOLDISTSTATS catalog table for the selected object</td>
</tr>
<tr>
<td>HISTORY</td>
<td>Displays statistics from the SYSIBM history tables for the selected object</td>
</tr>
<tr>
<td>INDEXSTATS</td>
<td>Displays statistical information from the SYSIBM.SYSINDEXSTATS catalog table for the selected object</td>
</tr>
<tr>
<td>PDISTSTATS</td>
<td>Displays statistics from the SYSIBM.SYSCOLDISTSTATS catalog table for the selected object</td>
</tr>
<tr>
<td>TABSTATS</td>
<td>Displays statistics from the SYSIBM.SYSTABSTATS catalog table for the selected object</td>
</tr>
</tbody>
</table>

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 70 on page 514 lists all of the codes and describes the type of list produced.

### Table 70: List commands

<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>Lists all the accelerators defined on the system</td>
</tr>
<tr>
<td>ACCPG</td>
<td>Lists all the accelerator packages</td>
</tr>
<tr>
<td>ACCTB</td>
<td>Lists all accelerator tables defined on an accelerator</td>
</tr>
<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>
<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><strong>Level-one list command</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------</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>
</tbody>
</table>
| DS                        | Lists the data sets  
  When used with the SYSPROC.ADMIN_DS_LIST stored procedure, the DS command can display data set information on a remote SSID. |
| DT                        | Lists the data types |
| ED                        | Invokes the options for the CATALOG MANAGER data editing function  
  To invoke the IBM DB2 data editor (if it is installed) and edit data from TB, VW, SY, and AL lists, uncomment the ED command in the CATALOG MANAGER commands table. |
<p>| EN                        | Lists the environments |
| FK                        | Lists the foreign keys |
| FN                        | Lists the routines (functions) |
| FO                        | Lists the routine (function) options |
| FP                        | Lists the function parameters |
| FS                        | Lists the SQL procedure sources |
| IC                        | Lists the image copies |
| IL                        | Lists the IP addresses for a given location |
| IM                        | Displays a mixed list with each index on one line, and each key column on a following line in key order |
| IN                        | Lists the location names and IP addresses of remote systems |
| IP                        | Lists the index partitions |
| IS                        | Lists the index spaces |
| ISS                       | Lists the index space statistics from the SYSIBM.SYINDEXSPACESTATS catalog table, which stores real-time statistics (RTS) |
| IX                        | Lists the indexes |
| IXC                       | Lists the time windows that specify index cleanup processing from the SYSIBM.INDEXLEANUP table |
| JB                        | Lists the objects from the installed JAR |
| JC                        | Lists the Java class information for the installed JAR |
| JP                        | Lists the build options that were used to build the installed JAR |
| JT                        | Lists the Java paths |
| KC                        | Lists the key columns |</p>
<table>
<thead>
<tr>
<th>Level-one list command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KT</td>
<td>Displays the contents of the SYSIBM.SYSKEYTARGETS catalog table, which contains key-targets of extended indexes</td>
</tr>
<tr>
<td>KTD</td>
<td>Displays the contents of the SYSIBM.SYSKEYTGTDIST catalog table, which contains distribution information for key-targets of extended index keys</td>
</tr>
<tr>
<td>KTDH</td>
<td>Displays the contents of the SYSIBM.SYSKEYTGTDIST_HIST catalog table, which contains history information for key-targets of extended index keys</td>
</tr>
<tr>
<td>KTDS</td>
<td>Displays the contents of the SYSIBM.SYSKEYTGTDISTSTATS catalog table, which contains the key-targets of data-partitioned secondary indexes</td>
</tr>
<tr>
<td>KTH</td>
<td>Displays the contents of the SYSIBM.SYSKEYTARGETS_HIST catalog table, which contains history information for key-targets of extended indexes</td>
</tr>
<tr>
<td>KTS</td>
<td>Displays the contents of the SYSIBM.SYSKEYTARGETSTATS catalog table, which contains partition statistics for key-targets of extended index keys</td>
</tr>
<tr>
<td>KU</td>
<td>Lists the constraint key columns</td>
</tr>
<tr>
<td>LI MX objectType qualifier</td>
<td>When issued from the Command line, displays a secondary list that shows multiple object types associated with certain source object types The object name must be fully qualified.</td>
</tr>
<tr>
<td>LIST</td>
<td>When entered in the Cmd column of a list line, produces a sublist qualified by the object name on the current list 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.</td>
</tr>
<tr>
<td>LK</td>
<td>Lists the limit keys</td>
</tr>
<tr>
<td>LL</td>
<td>Lists the logical unit (LU) names for a location from the SYSIBM.LULIST table</td>
</tr>
<tr>
<td>LM</td>
<td>Lists the limits for LUNAMEs and MODENAMEs from the SYSIBM.LUMODES table</td>
</tr>
<tr>
<td>LO</td>
<td>Lists the accessible remote locations from the SYSIBM.LOCATIONS table</td>
</tr>
<tr>
<td>LS</td>
<td>Lists the mode names for SQL requests from the SYSIBM.LUMODESELECT table</td>
</tr>
<tr>
<td>LU</td>
<td>Lists the remote locations that communicate with DB2 from the SYSIBM.LUNAMES table</td>
</tr>
<tr>
<td>MK</td>
<td>Lists the column masks from the SYSIBM.SYSControlS table</td>
</tr>
<tr>
<td>MQT</td>
<td>Lists the materialized query tables</td>
</tr>
<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>Level-one list command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>OS</td>
<td>Displays the contents of the SYSIBM.SYSLOBSTATS catalog table, which contains the statistics from LOB table spaces</td>
</tr>
<tr>
<td>PA</td>
<td>Lists the plan authorizations</td>
</tr>
<tr>
<td>PDD</td>
<td>Lists information about the objects that have pending changes to data definitions from the SYSIBM.SYSPENDINGDDL table</td>
</tr>
<tr>
<td>PG</td>
<td>Lists the packages</td>
</tr>
<tr>
<td></td>
<td>The PG command can be issued on a sequence object in a Sequences List. The dependency of the package on the sequence object must be specified in the SYSIBM.SYSPACKDEP catalog table.</td>
</tr>
<tr>
<td>PGC</td>
<td>Lists the package copies from the SYSIBM.SYSPACKCOPY table</td>
</tr>
<tr>
<td>PI</td>
<td>Lists the packages that a plan can use</td>
</tr>
<tr>
<td>PK</td>
<td>Lists the primary keys</td>
</tr>
<tr>
<td>PL</td>
<td>Lists the plans</td>
</tr>
<tr>
<td>PM</td>
<td>Lists the row permissions from the SYSIBM.SYSControlS table</td>
</tr>
<tr>
<td>PR</td>
<td>Lists the procedures</td>
</tr>
<tr>
<td>PT</td>
<td>Lists the partitions</td>
</tr>
<tr>
<td>QRO</td>
<td>Lists the optimization parameters for queries from the SYSIBM.SYSQUERYOPTS table</td>
</tr>
<tr>
<td>QRP</td>
<td>Lists the plan hint information for queries from the SYSIBM.SYSQUERYPLAN table</td>
</tr>
<tr>
<td>QRY</td>
<td>Lists the queries from the SYSIBM.SYSQUERY table</td>
</tr>
<tr>
<td>RD</td>
<td>Lists the object role dependencies</td>
</tr>
<tr>
<td>RE</td>
<td>Lists the referential constraints from the SYSIBM.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.SYSSEQUENCES table</td>
</tr>
<tr>
<td>SG</td>
<td>Lists the storage group objects</td>
</tr>
<tr>
<td>ST</td>
<td>Lists the strings</td>
</tr>
<tr>
<td>SU</td>
<td>Lists the system privileges for the AUTHID pattern that you entered</td>
</tr>
<tr>
<td>SY</td>
<td>Lists the synonyms</td>
</tr>
<tr>
<td>Level-one list command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SYNC</td>
<td>Lists the columns from the BMCSYNC table for the BMC utilities. The BMCSYNC table contains information about the status of the objects that the currently executing utilities are accessing. 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. The product lists the following CATALOG MANAGER object types:</td>
</tr>
<tr>
<td></td>
<td>■ AL—alias</td>
</tr>
<tr>
<td></td>
<td>■ TB—clone table, global temporary table, history table, XML implicit table, or table</td>
</tr>
<tr>
<td></td>
<td>■ MQT—materialized query table</td>
</tr>
<tr>
<td></td>
<td>■ VW—view</td>
</tr>
<tr>
<td></td>
<td>■ XT—auxiliary table</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>VAR</td>
<td>Lists the global variables from the SYSIBM.SYSVARIABLES table. <strong>Note:</strong> This command is not available from the main menu.</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 the COPY PLUS cabinet copies</td>
</tr>
<tr>
<td>Level-one list command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</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 71 on page 519* lists the user commands.

**Table 71: 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**

The following table 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 72: Command-line commands for browsing or editing data

<table>
<thead>
<tr>
<th>Command (short form)</th>
<th>Function E=Edit, B=Browse</th>
<th>Description</th>
</tr>
</thead>
</table>
| CANCEL               | E,B                       | Ends the edit or browse session  
In data editing mode, changes are not saved. |
| CHANGE               | E                         | Locates and modifies a character string that is specified in the command  
The syntax of the command is  
CHANGE 'targetString' 'replacementString'.  
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.  
**Note:** The :columnNumber identifier references the column number as defined in the table structure, not as displayed on the panel. |
| COPY                 | E                         | Inserts rows from a source table into the displayed target table |
| DOWN                 | E, B                      | In column view, scrolls forward through rows; in row view, scrolls forward through columns |
| EDIT                 | E, B                      | When the cursor is in a data field, opens an ISPF edit session in which you can change the data in the field |
| END (F3)             | E, B                      | Ends the edit or browse session  
In data editing mode, saves your changes and ends the edit session normally |
### Command (short form) | Function E=Edit, B=Browse | Description
---|---|---
FIND (F) | E, B | Locates a character string that you specify in the command. Usually, you enter the FIND command, followed by a space and then the character string. To locate a character string that includes blank characters, you must delimit the character string.

The FIND command searches for the specified string, starting at the top of the display or at the cursor position. If the character string is located, the result of a FIND command places the cursor at the start of the character string. If the character string is not located, either the *Bottom of Data reached* or *Top of Data reached* message is displayed.

The FIND command can search only in columns that can be displayed. Large character and graphic columns might not be displayed in their entirety, therefore a FIND command might not return occurrences of a character string embedded in such column types.

The following parameters can be used with the FIND command. The syntax of the command is FIND characterString PARAMETER.

- NEXT is the default parameter. Using the NEXT parameter returns the same result as using the FIND command with only the character string.

- PREV locates the first occurrence of the character string preceding the cursor location or the first line of the displayed data.

- FIRST searches the data for the first occurrence of the character string.

- LAST searches the data for the last occurrence of the character string.

- ALL searches the data for each occurrence of the character string, beginning at the first line and continuing to the last line.

- :columnIdentifier restricts the search to a single column. Valid column identifiers are :columnName, :columnNumber, and :columnLabel.

**Note:** The :columnNumber identifier references the column number as defined in the table structure, not as displayed on the panel.

The FIND command is valid only from the **Command** line.
<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>
<tr>
<td>IM</td>
<td>E, B</td>
<td>Displays the following information about a table or view:</td>
</tr>
<tr>
<td>IX</td>
<td></td>
<td>■ Qualified index name</td>
</tr>
<tr>
<td>KC</td>
<td></td>
<td>■ Cluster information</td>
</tr>
<tr>
<td>IX</td>
<td></td>
<td>■ Type of key</td>
</tr>
<tr>
<td>IX</td>
<td></td>
<td>■ Type of index</td>
</tr>
<tr>
<td>IX</td>
<td></td>
<td>■ Column name in the index</td>
</tr>
<tr>
<td>IX</td>
<td></td>
<td>■ Position of the column in the key</td>
</tr>
<tr>
<td>IX</td>
<td></td>
<td>■ Order of the column in the key</td>
</tr>
<tr>
<td>LEFT nnn</td>
<td>E, B</td>
<td>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.</td>
</tr>
<tr>
<td>LEFT M (F10)</td>
<td>E, B</td>
<td>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.</td>
</tr>
<tr>
<td>MORE</td>
<td>E, B</td>
<td>Displays additional rows when the address space is insufficient to display all of the selected rows.</td>
</tr>
<tr>
<td>OPT</td>
<td>E, B</td>
<td>Displays the Edit and Browse Options panel.</td>
</tr>
<tr>
<td>RCHANGE</td>
<td>E</td>
<td>Locates and modifies the next occurrence of the character string that is specified in the CHANGE command.</td>
</tr>
<tr>
<td>RESET RES</td>
<td>E, B</td>
<td>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.</td>
</tr>
<tr>
<td>RFIND (F5)</td>
<td>E, B</td>
<td>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.</td>
</tr>
<tr>
<td>Command (short form)</td>
<td>Function E=Edit, B=Browse</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>RIGHT ( nnn )</td>
<td>E, B</td>
<td>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.</td>
</tr>
<tr>
<td>RIGHT M (F11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROWVIEW ROW</td>
<td>E, B</td>
<td>Switches the display from column view to row view.</td>
</tr>
<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. You can also issue the SORT command followed by a column identifier and sort order indicator to perform the SORT on the editing panel, for example, SORT DATE ASC.</td>
</tr>
<tr>
<td>THAW</td>
<td>E, B</td>
<td>Reverses the action of the FREEZE command.</td>
</tr>
<tr>
<td>UNDO</td>
<td>E</td>
<td>In row view, reverses the changes made during an edit session.</td>
</tr>
<tr>
<td>UP</td>
<td>E, B</td>
<td>In column view, scrolls backward through rows; in row view, scrolls backward through columns.</td>
</tr>
<tr>
<td>ZOOM (F4)</td>
<td>E, B</td>
<td>Opens an ISPF session in which you can view the data in the field that is too large to display in CATALOG MANAGER. In data editing mode, you can also modify the data.</td>
</tr>
</tbody>
</table>

### Line commands

The following table 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>
<thead>
<tr>
<th>Command (short form)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I (^a)</td>
<td>Inserts one or more blank lines below the line on which the command is entered</td>
</tr>
<tr>
<td>D (^a) (^b)</td>
<td>Deletes one or more lines</td>
</tr>
<tr>
<td>Command (short form)</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>R a b</td>
<td>Inserts a line with the same values as the line on which the command is entered</td>
</tr>
<tr>
<td>C a b</td>
<td>Identifies the source line of a copy operation</td>
</tr>
<tr>
<td>M a b</td>
<td>Identifies the source line of a move operation</td>
</tr>
<tr>
<td>A</td>
<td>Indicates the line after which the copied or moved data will be inserted</td>
</tr>
<tr>
<td>B</td>
<td>Indicates the line before which the copied or moved data will be inserted</td>
</tr>
<tr>
<td>U a b</td>
<td>Reverses the changes that you have made since the most recent SAVE</td>
</tr>
<tr>
<td>Z V !</td>
<td>Displays the data in the row on which the command is entered vertically rather than horizontally, with one column per line. This vertical row format is called row view. By using row view, you can display columns that are too large to display in column view. To return the display to column view, press END.</td>
</tr>
</tbody>
</table>

- You can follow these commands with a numeric value to apply the command multiple times.
- These commands can be used with block identifiers. For example, you can enter DD on two lines to mark those lines and the lines between them for deletion.
Glossary

A

ACM

The product code that BMC uses to identify the CHANGE MANAGER product.

ACT

The product code that BMC uses to identify the CATALOG MANAGER product.

action code

A one-character or two-character command that you can enter on one of the lines of a list panel. Although you can only enter one command per line, you can enter multiple commands on a single panel.

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 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.
AEXPIN

The ddname of the input stream that the Execution component uses to control parallel processing.

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.

AEXPRnnn

The ddname of one of the initiator outputs of the Execution component. When the worklist parallelism feature of the Database Administration solution is used, this output data set contains AEXPRINT output from each BMC Cross-System Image Manager (XIM) initiator.

AEXPTRAC

The ddname of the trace output of the Execution component. When the worklist parallelism feature of the Database Administration solution is used, this output data set contains tracing records.

AEXSYnnn

The ddname of one of the initiator outputs of the Execution component. When the worklist parallelism feature of the Database Administration solution is used, this output data set contains system messages and job information for each BMC Cross-System Image Manager (XIM) initiator.

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.BMCSLIB($AJXDOCV).

AJXIN

The ddname of the input stream that the Batch Execution JCL Generation component uses.
AJXPOFIN

The ddname of the override input stream that the Batch Execution JCL Generation component uses.

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 installation options 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.

alter-type work ID

A work ID that ALTER and CHANGE MANAGER use to perform modifications to the local DB2 subsystem.

ALU

The product code that BMC uses to identify the ALTER product.

ALUIN

The ddname of the input stream that the Import, Baseline, Baseline Report, Compare, Analysis, and CM/PILOT components of CHANGE MANAGER use.
ALUPRINT

The ddname of the diagnostic output of the Import, Baseline, Baseline Report, Compare, and Analysis components of CHANGE MANAGER.

Analysis

A component of ALTER and CHANGE MANAGER that reads the change and migration requests for a particular work ID and generates a worklist to implement the requests. Analysis checks for consistency with the DB2 catalog before it builds the worklist.

Application

In the CM/PILOT component, the association of a group of CHANGE MANAGER profiles that are used repeatedly for the change management tasks of a specific DB2 application.

ASU

The product code that BMC uses to identify the DASD MANAGER PLUS product.

attribute

A value that defines certain properties of an object. Each attribute of an object can occur only once and has a single value from a finite list of possible values. For example, some of the attributes of the TABLE object are database, table space, and EDITPROC. Attributes differ from sub-elements because sub-elements can occur multiple times and can have attributes of their own. For example, columns are sub-elements of tables.

AUTHID

See “authorization ID” on page 528.

authorization ID

An identifier that is allowed a set of privileges. An example of the authorization ID is the owner of a table space, database, storage group, or synonym. An authorization ID is the implicit qualifier of a table, view, alias, or index name.

auxiliary index
An index on an auxiliary table. Each index refers to a large object (LOB) column.

auxiliary list

A list of several auxiliary objects (such as auxiliary table spaces, tables, or indexes) of only one type.

auxiliary table

A table that contains a single large object (LOB) column. An auxiliary table resides in an auxiliary (or LOB) table space.

auxiliary table space

A nonpartitioned table space that contains the data for a large object (LOB) column in an auxiliary table.

B

base table

A table that contains a ROWID column and the definition for a large object (LOB) column. A base table is incomplete if it does not contain any auxiliary objects, such as auxiliary table spaces, tables, and indexes.

base table space

A table space that contains base tables.

baseline

A component of CHANGE MANAGER that captures a set of DB2 structure definitions from either the DB2 catalog or a DDL file at a specific point in time.

baseline name template

A template that a baseline profile can contain. This template is used to create the names of the baselines that are established with the profile. The template can include arbitrary text and the special character sequences #### and @@@@@@ or @@@@@@@@, or a combination of both types of sequences. When the baseline is created, ascending numbers replace the ####
sequence and the current date replaces the @@@@@@@@ or @@@@@@@@@@@@ sequence. For @@@@@@@@, the date is in the form YYMMDD. For @@@@@@@@@@@@, the date is in the form YYYYMMDD.

baseline profile

A BMC object that contains the information that is necessary to establish a baseline. The two types of baseline profiles are catalog and DDL.

batch component

A component that can run in batch mode. Batch components of CHANGE MANAGER include Analysis, Baseline, Baseline Report, Compare, Execution, and Import.

BMC object

A logical entity that contains data that is necessary for performing tasks. BMC objects have a two-part name \((a,b)\). The BMC objects for CHANGE MANAGER include work IDs, worklists, unload data sets, baselines, profiles, CDL files, DDL files, internal tables, task IDs, CM/PILOT worklists, applications, and script tables.

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.

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 Administrative products that allows them to access the DB2 catalog indirectly when making information queries. Catalog indirection is accomplished by using aliases that point to a copy of the DB2 catalog. The major benefit is to reduce catalog contention.

catalog baseline

A baseline that is established on part of the DB2 catalog. A catalog baseline must include a scope, which might be defined directly in a baseline profile or by reference to a scope in a migrate profile.

CATALOG MANAGER for DB2

A tool that is designed to automate 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, re-create dropped structures and data, and browse and edit table data.

CDL

See “Change Definition Language (CDL)” on page 531.

Change Definition Language (CDL)

A BMC proprietary language that is used to specify changes to DB2 data structures.
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.

change rule

An element of a migrate-type work ID or a migrate profile. Change rules define the automatic object attribute changes that are made during a migration or change migration process. These rules can include changes, or sub-element inclusion or exclusion. Change rules can also force VCAT definitions for table spaces and indexes.

checkpoint

A point at which information about the status of a job and the system can be recorded so that the job step can later be restarted.

See also “sync point” on page 550.

child

A DB2 object that contains the foreign keys which reference the primary key in a parent table.

See also “parent” on page 545.

CLIST

Command list.

CM/PILOT for DB2

A component of CHANGE MANAGER that automates the DB2 change management processes that you perform using CHANGE MANAGER.

CM/PILOT worklist

A data set that contains the ordered commands, keywords, and parameters that CHANGE MANAGER needs to process a task ID.
command

A token that you can enter at the command prompt on a panel.

*See also* “action code” on page 525.

**commit**

An operation that terminates a unit of work. A commit releases all locks. Data that was changed is now consistent.

**Compare**

A component of CHANGE MANAGER that identifies the differences between two sets of data structures and then generates a CDL file. You can review the differences and decide how to apply the differences to implement the necessary changes. You can compare data structures that are stored in a DDL file, baseline, worklist, or DB2 catalog.

**Compare1**

The primary input to the compare process.

*See also* “source” on page 548.

**Compare2**

The secondary input to the compare process.

*See also* “target” on page 550.

**component**

A major functional unit of ALTER or CHANGE MANAGER, such as Analysis, Execution, Specification, or Import.

**constraint**

*See* “referential constraint” on page 547.
Cross-System Image Manager (XIM)

A BMC technology that provides sysplex performance improvements by enabling the distribution and management of discrete units of work (UOW) across one or more OS/390 and z/OS systems. The BMC products that use XIM can divide single, long-running tasks into multiple parallel tasks that can be run across multiple computers in the sysplex, thus decreasing the overall elapsed time. The products can also be used with XIM in a data sharing environment on a single OS/390 or z/OS image.

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 Manipulation Language (DML)

In the CM/PILOT component of CHANGE MANAGER, SQL-like statements that can be used to update, delete, and migrate data structures.

data set sizing

The process of determining data set allocations, especially as used by the JCL Generation component. Data set sizing is distinct from space estimation.
data structure

An object that is defined in the DB2 catalog. Objects include storage groups, databases, table spaces, tables, indexes, foreign keys, views, synonyms, aliases, and triggers.

database administrator (DBA)

An individual who is responsible for the design, development, operation, security, maintenance, and use of databases.

database request module (DBRM)

A module that contains SQL statements which the DB2 precompiler has extracted from a source program.

DB2 catalog

System tables, maintained and used by DB2, that contain descriptions of DB2 objects such as tables, views, and indexes.

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 (-).

DBA

See “database administrator (DBA)” on page 535.

DBCS

See “double-byte character set (DBCS)” on page 537.

DD statement

Data Definition statement.

DDL
See “data definition language (DDL)” on page 534.

**DDL baseline**

A baseline that is established on a file that contains DDL.

**ddname**

See “data definition name (ddname)” on page 534.

**default options module (DOPTS)**

See “installation options module” on page 540.

**default value**

A predetermined value, attribute, or option that is assumed when no other is explicitly specified.

**delimited identifier**

An SQL identifier that is enclosed within escape characters.

**dependencies**

The name or values of objects which another object uses as part of its definition or as a hierarchical subordinate.

**dependent**

A child object (row or table) that has at least one parent.

**dependent object**

An object whose definition relies on the name or the values of another object. The dependent object references the other object.

**destination**
The intended receiving location for CDL or a worklist.

device type

The type of disk device used for DB2 data set allocation, such as 3380 and 3390, or generic types TAPE and CART.

DOPTS

See “installation options module” on page 540.

double-byte character set (DBCS)

A delimited set of characters in which each character is represented by two bytes. Katakana and other lowercase characters are nonstandard characters and must be contained within double quotes.

duplicate

An action that you can make on a database object. You can duplicate single or multiple objects within the same database system. If you want to create an object that is like an existing object, you can use the Create Like command to duplicate the existing object. You can then change the object name and make any other necessary changes.

E

edit procedure

See “EDITPROC” on page 537.

EDITPROC

An edit procedure that defines an editing routine to be invoked just after a record that corresponds to a table row is retrieved and just before that record is stored. Editing routines allow for data compression, decompression, and encrypting.
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 your 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.

**Execution**

A component of ALTER and CHANGE MANAGER that carries out the commands in a worklist.

**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).

**F**

**field procedure**

*See “FIELDPROC” on page 538.*

**FIELDPROC**

A user-written exit routine that is designed to receive a single value and transform (encode or decode) it in any way that the user specifies.

**foreground component**

A component that can run in foreground mode. Foreground components in CHANGE MANAGER include Front End, Specification, Analysis, Compare, JCL Generation, and Import.

**Front End**

A component of ALTER and CHANGE MANAGER that acts as the interface between the user and the other components. Front End is an interactive ISPF dialog that is responsible for creating and maintaining BMC objects and facilitating the generation of JCL.
full-recovery baseline

A baseline that captures data and the data structure definitions at a specific point in time.

G

GDG

See “generation data group (GDG)” on page 539.

generation data group (GDG)

A finite number of data sets that are kept in chronological order. Each data set is a generation data set.

Group ID

In an outbound migrate profile, a four-character identifier that links locations (or application instances) together. If a migrate profile defines one or more Group IDs, Analysis and Compare generate a single output file (worklist or CDL) for each group. If groups are not defined, Analysis and Compare generate one output file for each location. Locations within a group must reside on the same physical DB2 subsystem.

H

hash value

A number that appears at the end of commands in worklists or CDL files. The hash value is generated based on the contents of the command line and allows the products to determine whether the line has been manually modified since the file was generated.

HLQ

High-level qualifier of a data set.
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. You can make an image copy of an index.

Import

A component of CHANGE MANAGER that converts statements that are stored in a CDL, DDL, or DML file into change requests in a work ID. In ALTER, Import converts statements that are stored in a DDL file.

import or importation

The process of obtaining an object or an object-set definition from a file or an external database and applying it to an alter-type work ID.

inbound migrate profile

A profile that can be used with the Import component to automate changes to object attributes. An inbound migrate profile can contain only change rules (no locations or scope).

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 module

An assembler module that contains keywords whose global values determine the operating environment for a BMC product.

internal table

A table that ALTER or CHANGE MANAGER use to store information.
ISPF skeletons

Data definition statement templates that JCL Generation uses. The skeletons are described in HLQ.BMCSLIB($AJXDOC).

J

JCL

Job Control Language that is used to execute processes in batch mode.

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 ALTER and CHANGE MANAGER that constructs a job control language (JCL) file for running the components in batch mode. When you choose to build JCL, JCLGEN is passed to the worklist that contains the control statements. ALTER and CHANGE MANAGER resolve all data set names that are entered with symbolic variables on the interface panels. JCLGEN resolves all data sets that are passed from the option panels and the unload data sets that are used by the Execution facility. The generated JCL includes data definition statements (ddnames) for all data sets that are needed by Execution, as well as the EXEC statement for the program and any necessary control parameters.

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.

JCLGEN

See “JCL Generation (JCLGEN)” on page 541.

job

A batch unit of work that is defined by JCL, a work ID, and a worklist to perform tasks. ALTER and CHANGE MANAGER use the information that you supply for a work ID to generate the
worklist, which provides the BMC JCL Generation component (JCLGEN) with the information that is necessary to build the JCL to run a job.

L

large object (LOB) column

A type of column that is used to store large objects (LOBs), such as images, audio, video, text, or graphics, as strings. The data type of the column is defined as LOB (such as a binary large object, or BLOB; character large object, or CLOB; or double-byte character large object, or DBCLOB) or as a distinct or user-defined type (UDT) that is based on a LOB data type. A LOB column resides in an auxiliary table.

large object (LOB) table space

See “auxiliary table space” on page 529.

LINK library

A partitioned, cataloged data set that is used to store and retrieve all or part of a program in a form that is suitable for loading into main storage for execution. The LINK library contains executable modules that perform a product’s processes.

LOB DATA MOVER

In the Database Administration solution, a program that is used to unload and load data that is contained in large object (LOB) columns.

location

An arbitrary identifier that groups change rules for a particular destination. An outbound migrate profile can define one or more locations. A location is not a DB2 location identifier.

M

menu

A list of action options. 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.
migrate

The process of moving DB2 data structures, data structures and data, or data structure changes, from an origin to a destination subsystem.

migrate profile

A profile that can be reused to select a set of objects, to customize changes to objects which are migrated to different locations or used in a comparison, or both.

migrate-type work ID

A work ID that is used to migrate data structures, data, or both. Worklists that are generated from a migrate-type work ID do not contain SQL DROP statements. These work IDs can have migrate options and change rules.

mixed list

A panel in the Specification component that can display multiple DB2 object types.

N

name template

See “baseline name template” on page 529 and “work ID name template” on page 553.

name propagation

The process of extending to dependent objects the changes that you make in a referenced object. For example, if you change the name of a table that is referenced in a view, name propagation replicates the new name in the view definition.

cnull

A special value that indicates the absence of information.
object

A DB2 entity that can be created or dropped. Objects are storage groups, databases, table spaces, tables, indexes, foreign keys, views, synonyms, aliases, or triggers. Unlike DB2, which treats foreign keys and check constraints as table attributes, ALTER and CHANGE MANAGER treat foreign keys and check constraints as independent objects.

object list

A list of one or more objects of different types.

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 ALUIN, AEXIN, or AJXIN input stream.

origin

The sending location or source of a migration or a change migration.

orphaned auxiliary index

An auxiliary index that is not associated with an existing auxiliary table.

orphaned auxiliary table space

An auxiliary table space that does not contain an auxiliary table.

outbound migrate profile

A profile that is used in performing a change migration. An outbound migrate profile might define one or more locations that contain change rules that are defined for any or all of those locations. Optionally, an outbound migrate profile can have a scope to select the DB2 objects on which to operate.
page

A unit of storage within a table space (4K or 32K) or index space (4K). A page in a table space contains one or more rows of a table. 8K and 16K pages can be used.

parent

A DB2 object that contains the primary key which might be referenced by one or more foreign keys in the child table.

See also “child” on page 532.

partitioned data set (PDS)

A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data. Synonymous with program library.

partitioned table space

A table space that is subdivided into parts (based upon index key range), each of which can be processed independently by utilities.

pattern

A rule that is applied to the naming of objects of a specified type. You can use wildcard characters (% and *) when you define a pattern.

PDS

See “partitioned data set (PDS)” on page 545.

privilege

The capability of performing a specific function (authorization) on an object. Privileges might be explicitly or implicitly granted.
profile

A collection of scope rules, change rules, and locations that enables you to define and control a change, migrate, or baseline process.

See also “baseline profile” on page 530, “migrate profile” on page 543, “inbound migrate profile” on page 540, and “outbound migrate profile” on page 544.

protected baseline

A baseline that is designated as protected from deletion. A protected baseline cannot be deleted until the protected designation is removed.

See also “unprotected baseline” on page 551.

receive-type work ID

A work ID that is used to create new data structures and load migrated data on a different subsystem.

recovery

The process of restoring a set of data structure definitions to their state at a particular point in time. Recovery involves comparing the DB2 catalog to a baseline, importing the CDL, generating a worklist with Analysis, and executing the worklist. If the baseline is a full-recovery baseline, you can recover data and the data structure definitions.

reference location

A location in a profile whose rules are used by another location. For example, a location called Houston might have that rules that are explicitly defined, while locations called Austin and Dallas might use Houston as a reference location.

reference profile

A profile whose scope is used by another profile. A baseline profile can reference a migrate profile, and a migrate profile can reference a baseline profile. Using a reference profile enables you to define the scope only once and thus eliminates errors that might arise from redesigning the scope.
referenced object

An object that a dependent object references. If you change the definition of a referenced object, dependent objects might not continue to function properly.

referential constraint

The requirement that nonnull values of a designated foreign key are valid only if they equal values of the primary key of a designated table. The relationship between the primary key in the parent table and a foreign key in a dependent table is used to establish referential integrity in a database. A referential constraint is always assigned a name to distinguish it from other constraints.

referential integrity

The condition that exists when all intended references from data in one column of a table to data in another column of the same or a different table are valid. Maintaining referential integrity requires enforcing referential constraints on all LOAD, RECOVER, INSERT, UPDATE, and DELETE operations.

S

SBCS

See “single-byte character set (SBCS)” on page 548.

scope

The final set of DB2 objects that are selected from the catalog on which the product operates. A scope includes objects that are explicitly selected and any of their object dependencies. For the Baseline component, the scope selects the objects that are captured in the baseline. For the Compare and Analysis components of CHANGE MANAGER, the scope selects the objects that are included in the comparison or analysis process. A scope consists of one or more scope rules.

scope rule

A specification for selecting DB2 objects from the catalog by object type and name.

Script table
In the CM/PILOT component of CHANGE MANAGER, ordered steps that prompt you for the information that is required to perform a change management task.

**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.

**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.

**single-byte character set (SBCS)**

A character set in which each character is represented by a one-byte code.

**source**

In CHANGE MANAGER, the original object of a migration or the original object of a Compare process. When migrating objects or databases, the source database is the database from which you are migrating. The Compare process compares a source data structure with a target data structure. This process synchronizes two data structures, and the source is the data structure that needs to change.

*See also “Compare1” on page 533.*

**Space Estimation**

A feature of ALTER and CHANGE MANAGER that enables you to determine the amount of space that a table space or index will require, based on the object definitions and their estimated usages.

**Specification**

A component of ALTER and CHANGE MANAGER that enables you to create or edit data structure change or migration requests. Specification stores its change or migration requests in a work ID.
SQL

See “Structured Query Language (SQL)” on page 549.

SQLID

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.

structure-only baseline

A baseline that contains only data structure definitions. No data from those data structures are included.

Structured Query Language (SQL)

An ANSI-standard language for database definition, manipulation, and query.

sub-element

A component of an object. For example, a column is a sub-element of a table, and a volume is a sub-element of a storage group.

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 installation options modules, a symbolic variable should be preceded with two ampersands (&&). Symbolic variables should not be confused with global job variables (AJX-type), which have their values set for all users and all sessions.

See also “JCL Generation (JCLGEN)” on page 541.
SYNC

A worklist command that invokes a checkpoint to use for restart processing and commits data to DB2.

sync point

A completion flag that is set during the execution of a worklist. The Execution program writes sync points to the SYNC table whenever it encounters -SYNC or -STOP commands in the worklist input stream. All SQL statements between sync points are executed as a single DB2 transaction. If a worklist is halted before completion for any reason, sync points enable you to begin processing the worklist from the last sync point.

synchronization

The process of identifying structural differences between two copies of the same data structure and then making the data structures identical. For example, separate groups of developers might be independently modifying several copies of a set of data structures. At various times, the copies need to be synchronized to ensure that all of the developers are using the same structure definitions.

T

target

The object of a migration or the object of a Compare process. When migrating objects or databases, the target database is the database to which you are migrating. The Compare process compares a source data structure with a target data structure. This process synchronizes two data structures, and the target is the data structure that contains the wanted changes.

See also “Compare2” on page 533.

task ID

A unit of work in the CM/PILOT component.

template

A method of obtaining specifications for the definition of an auxiliary table space, table, or index. This definition can be replicated for each of the partitions in the base table space.
See also “baseline name template” on page 529 and “work ID name template” on page 553.

U

unit

A specific device, device type, or group of devices that are used in data set allocation.

unload data set

An object that is used to store data while DB2 objects are dropped and rebuilt. The unload data set is also referred to as a SYSREC data set.

unprotected baseline

A baseline that is not designated as protected from deletion.

See also “protected baseline” on page 546.

UOW

Unit of Work. A unit of work consists of the worklist commands that are bounded by the -BEGU and -ENDU commands and that are run in a BMC Cross-System Image Manager (XIM) initiator.

UOWTRnnnn

The ddname of the unit of work (UOW) output of the Execution component. When the worklist parallelism feature of the Database Administration solution is used, this output data set contains tracing records for each BMC Cross-System Image Manager (XIM) initiator.

user options

A set of options that are stored in the user’s profile and that are used by Front End and JCL Generation for running ALTER or CHANGE MANAGER components. The user options are initially set from the installation options module the first time that the user runs ALTER or CHANGE MANAGER.
V

variable

See “symbolic variable” on page 549.

VCAT allocation

A volume placement parameter that specifies the data set high-level qualifier that is appropriate for the DB2 subsystem. Use this field for nonstorage group allocation only.

versioning

The process of comparing baselines that represent the same set of structures at different points in time in order to change a version of a data structure.

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:

- The % and * represent any character string.
- The _ and ? represent a single character.

work ID

A unit of work with a two-part name (owner.name) that contains change or migration requests in change definition (CD) tables. The change or migration requests can be either imported or created manually through the Specification component.

work ID name

A work ID name is a string of up to eighteen alphanumeric characters excluding percent (%), asterisk (*), underscore (_), and space. When you specify a work ID name, you can use a wildcard pattern to display a group of similar names.
work ID name template

A template from which the name of a work ID is created. You can use this template to create or replace a work ID in batch mode. The name template contains the special character sequences #### and @@@@@@@ or @@@@@@@@@, in addition to text characters. An ascending sequence of numbers replaces the #### sequence, and the current date replaces the @@@@@@@ or @@@@@@@@@ sequence when the work ID is generated. You can create and replace receive-type work IDs (using name templates) with the Import and Execution components.

work ID owner

The Authorization ID of the creator of the work ID.

worklist

A data set that contains commands for implementing a data structure change or migration.

worklist execution log

See “AEXPRINT” on page 526.

worklist parallelism

A feature in the Database Administration solution that reduces the elapsed time for executing a worklist that is generated by the CHANGE MANAGER product.

X

XIM

See “Cross-System Image Manager (XIM)” on page 534.

XIM initiator

A program that executes one or more units of work (UOW).
Index

Symbols

- BMCU worklist command 495
- DBUG worklist command 495
- DSN1 worklist command 496
- MERG worklist command 496
- MODI worklist command 497
- NOOP worklist command 497
- QUI worklist command 498
- REPO worklist command 498
- REPX worklist command 499
- SQLM worklist command 499
- STOS worklist command 500

! line command 523
? command 512
(DB2) Connect saved connections 166

**PREFIX** TEMPLATE descriptor variable
  PREFIX symbolic variable 393
  SYSUID symbolic variable 395
  UID symbolic variable 396
  USERID symbolic variable 396
  ZPREFIX symbolic variable 398
  ZSYSID symbolic variable 398
  ZUSER symbolic variable 398

*DROP marker 264
*PERSIST option 76
&ACTCOLID ISPF variable 149
&ACTSRVR ISPF variable 149
&CLIST parameter 143
&DB2MAX parameter 143
&DB2MIN parameter 143
&HELP parameter 143
&LOAD parameter 143
&LOG parameter 143
&LSTO parameter 143
&NLIST parameter 143
&NOSERVER parameter 143
&NUM parameter 143

&OBJECTS parameter 143
&PARSE parameter 143
&PLAN parameter 143
&SSID ISPF variable 149
&VCAT ISPF variable 149
&WFEK parameter 143
+ command 512
+1 OUTPUT descriptor variable 388
+1 TEMPLATE descriptor variable 388
= command 512
=X command 44
$ACTCMD macro 143
$ACTEXC macro 147
$ACTLISTC macro 147
$ACTSQLD options 411
$ACTULOG macro 147
$ACTVARS macro 147

Numeric

10-byte object type address (TYPE) 147
16-byte function address (FUNC) 147
18-byte object name address (NAME) 147
2MEGSQL AEXIN keyword 377
2MEGSQL POF keyword 424
2SQL command 268
2WL command 502
8-byte object name qualifier (QUAL) 147

A

A line command 523
above-the-bar storage 66
ACC command 514
ACCEL command 502
ACCPG command 514
ACCTB command 514
ACM
definition 525
ACM AEXIN keyword 377
ACM_AMS POF keyword 424
ACM_ANALYSIS_SYSTOUT POF keyword 424
ACM_BASDIAG POF keyword 424
ACM_BRPTDIAG POF keyword 424
ACM_BRPTDSN POF keyword 424
ACM_CDLDASN POF keyword 425
ACM_CDLPS POF keyword 425
ACM_CDLSS POF keyword 425
ACM_CDLU POF keyword 425
ACM_CMPDIAG POF keyword 425
ACM_CPLCDLO POF keyword 425
ACM_CPLDIAG POF keyword 425
ACM_CPLWDSN POF keyword 426
ACM_CPLWDSNO POF keyword 426
ACM_DBRM1 POF keyword 426
ACM_DBRM2 POF keyword 426
ACM_DBRM3 POF keyword 426
ACM_DYNSORTW_NUM POF keyword 426
ACM_DYNSORTW_UNIT POF keyword 426
ACM_GLID POF keyword 426
ACM_IBMR_MAP_REQ POF keyword 427
ACM_IMPDIAG POF keyword 427
ACM_JDSN POF keyword 427
ACM_JDSNB POF keyword 427
ACM_JDSNBG POF keyword 427
ACM_JDSNBR POF keyword 428
ACM_JDSNC POF keyword 428
ACM_JDSNCPL POF keyword 428
ACM_JDSNPCLO POF keyword 428
ACM_JDSNE POF keyword 428
ACM_JDSNI POF keyword 428
ACM_PARALLEL_MAXINIT POF keyword 429
ACM_PARALLEL_MININIT POF keyword 429
ACM_PARALLEL_WORKLST POF keyword 429
ACM_PARALLEL_XIMGRP POF keyword 429
ACM_PARALLEL_XIMPROC POF keyword 429
ACM_PARALLEL_XIMSTRT POF keyword 430
ACM_PARALLEL_XIMTRCE POF keyword 430
ACM_PIC POF keyword 430
ACM_SAFTS POF keyword 430
ACM_SDSN POF keyword 430
ACM_SDSNE POF keyword 430
ACM_WDSL POF keyword 431
ACM_WLORDER POF keyword 431
ACM_WLORDERMSG POF keyword 431
ACM_WLPS POF keyword 431
ACM_WLSS POF keyword 431
ACM_WLU POF keyword 432
ACT
  definition 525
ACM product code 64
ACTCOMND member 138, 147
ACTCOMNU member 147
action code
  definition 525
ACTIVATE command 506
ACTPRSS CLIST 367
ACTUSER member 147
ACTVRDB Bind/Rebind plan 64
ACTVRDE Data Editor plan 64
ACTVRDG Generate SQL plan 64
ACTVRDH Utility Status Display plan 64
ACTVRDK Command Generation and Execution plan 64
ACTVRDL Log Table maintenance plan 64
ACTVRDM Display DB2 Catalog plan 64
ACTVRDS Search plan 64
ACTVRDU Grant Authorities plan 64
adding commands 139
ADDLOAD1 POF keyword 432
ADDLOAD2 POF keyword 432
ADMIN_COMMAND_DB2 stored procedure 160,
  165
ADMIN_DS_LIST stored procedure 160
Administrative Assistant for DB2 26
Administrative Products for DB2
  definition 525
AEXIN
  definition 525
AEXIN input stream
  user options 68
AEXIN keywords, list of 377
AEXPIN
  definition 526
AEXPRT
  definition 526
AEXPRnnn
  definition 526
AEXPTRAC
  definition 526
AEXSYnnn
  definition 526
AJX variables
  definition 526
AJX4PART SLIB variable 393
AJX4YDDD SLIB variable
DATEJ symbolic variable 387
JDATE symbolic variable 389
JU symbolic variable 391
JULAY symbolic variable 391
YE symbolic variable 398
YEAR symbolic variable 398
YYYYDDD symbolic variable 398
AJX5PART SLIB variable 393
AJXBMCCP SLIB variable 387
AJXCR SLIB variable
   CR symbolic variable 387
   IXCR symbolic variable 389
AJXDB SLIB variable 388
AJXDB2V2 SLIB variable 388
AJXDB2V3 SLIB variable 388
AJXDDN SLIB variable 389
AJXDDOPT SLIB variable 388
AJXDDQSC SLIB variable
   DDSEQ symbolic variable 388
   SEQ symbolic variable 394
   SQ symbolic variable 394
AJXDSN44 SLIB variable 387
AJXFCDM SLIB variable 388
AJXFJCHR SLIB variable 390
AJXGDGPC SLIB variable 388
AJXHM SLIB variable 388
AJXHMS SLIB variable
   HMS symbolic variable 388
   HO symbolic variable 389
   HOUR symbolic variable 389
   JHMS symbolic variable 389
   MI symbolic variable 392
   MINUTE symbolic variable 392
   SC symbolic variable 394
   SEC symbolic variable 394
   SECOND symbolic variable 394
   TI symbolic variable 395
   TIME symbolic variable 395
   TIME4 symbolic variable 395
AJXIC SLIB variable 389
AJXIN
   definition 526
AJXIN input stream 68, 355
AJXIX SLIB variable 389
AJXIXNOD SLIB variable 389
AJXIXSPC SLIB variable 389
AJXJAI UN SLIB variable 387
AJXJDDN SLIB variable 388
AJXJJULD SLIB variable 389
AJXJOBNM SLIB variable 390
AJXJOBT SLIB variable 390
AJXJPCOD SLIB variable 390
AJXJQID SLIB variable
   JQID symbolic variable 390
   WKID symbolic variable 397
   WORKID symbolic variable 397
AJXJSSID SLIB variable
   ATTACH symbolic variable 387
   GRPNM symbolic variable 388
   SS symbolic variable 394
AJXJYMD SLIB variable 391
AJXLDEFN SLIB variable 391
AJXLLQ SLIB variable 391
AJXLR SLIB variable
   LOCREM symbolic variable 391
   LR symbolic variable 391
   TYPE symbolic variable 396
AJXMEMBER SLIB variable 392
AJXMEMBR SLIB variable 392
AJXMESSID SLIB variable 392
AJXOBJT SLIB variable 393
AJXOBNAM SLIB variable 392
AJXOBNOD SLIB variable 393
AJXOBT SLIB variable 392
AJXODS44 symbolic variable 387
AJXPARTC SLIB variable
   DSNUM symbolic variable 388
   LDSNUM symbolic variable 391
   LPART symbolic variable 391
   PA symbolic variable 393
   PART symbolic variable 393
AJXPB SLIB variable 393
AJXPCMR SLIB variable 393
AJXPODAT edit macro 129
AJXPOFER message file 132
AJXPOFIN
   definition 527
AJXPOFIN input stream 355
AJXPOFVL
   definition 527
AJXPOFVL message file 132
AJXPRINT
   definition 527
AJXRHLQ SLIB variable 393
AJXRSEQ# SLIB variable 393
AJXRUNTP SLIB variable 394
AJXSEQ# SLIB variable 394
AJXSPNAM SLIB variable
SN symbolic variable 394
SPNAME symbolic variable 394
TSIX symbolic variable 395
AJXSSID SLIB variable
  JSSID symbolic variable 391
  SSID symbolic variable 394
AJXSTEPC SLIB variable 394
AJXSTEPN SLIB variable 394
AJXSYSM SLIB variable 395
AJXTBCRE SLIB variable 395
AJXTBNAM SLIB variable 395
AJXTBNOD SLIB variable 395
AJXTSLIB variable 395
AJXTS SLIB variable 395
AJXTSCR SLIB variable 395
AJXTSID SLIB variable 395
AJXTU1 SLIB variable 395
AJXTU2 SLIB variable 395
AJXTU3 SLIB variable 396
AJXUCMD SLIB variable 396
AJXUDOPT SLIB variable 396
AJXULLQ SLIB variable 396
AJXUPART SLIB variable 396
AJXUTID SLIB variable 396
AJXUVR1 SLIB variable 397
AJXUVR2 SLIB variable 397
AJXUVR3 SLIB variable 397
AJXUVR4 SLIB variable 397
AJXUVR5 SLIB variable 397
AJXVCAT SLIB variable 397
AJXWKID SLIB variable
  JOBNAME symbolic variable 390
  WORKID8 symbolic variable 398
AJXWKOWN SLIB variable 397
AJXYMD SLIB variable
  DA symbolic variable 387
  DATE symbolic variable 387
  DAY symbolic variable 387
  DT symbolic variable 388
  MO symbolic variable 392
  MONTH symbolic variable 392
  YMD symbolic variable 398
AJXYYDDD SLIB variable
  DDD symbolic variable 388
  JD symbolic variable 389
  JDAY symbolic variable 389
  JULIAN symbolic variable 391
  YY symbolic variable 398
  YYYYDDD symbolic variable 398
AL command 514
AL object type 36, 176, 178
alias object type 36, 178
ALID symbolic variable 387
ALL keyword 55, 177
ALLC installation option 404
alloc unit
  definition 527
ALTER command 506
ALTER for DB2
  definition 527
alter-type work ID
  definition 527
ALTERID AEXIN keyword 377
ALU
  definition 527
ALU AEXIN keyword 377
ALUIN
  definition 527
ALUPRINT
  definition 528
Analysis
  definition 528
ANALYZE command 367, 373, 502
AOPTS installation option 404
APO command 514
Application
  definition 528
Application ID 70
APPLY command 209, 502
ARCH_DATACLASS POF keyword 432
ARCH_DATACLASS_ALT POF keyword 432
ARCH_EXPDT 432
ARCH_MGMTCLASS POF keyword 432
ARCH_MGMTCLASS_ALT POF keyword 432
ARCH_PREFIX POF keyword 433
ARCH_PRIQTY 433
ARCH_RETPD POF keyword 433
ARCH_SECQTY 433
ARCH_STACK POF keyword 433
ARCH_STORCLASS POF keyword 433
ARCH_STORCLASS_ALT POF keyword 433
ARCH_THREASH POF keyword 433
ARCH_UNIT POF keyword 433
ARCH_UNIT_ALT POF keyword 434
ARH command 514
ASU
  definition 528
ASU AEXIN keyword 377
ASU_XP_LOGD_DATAC POF keyword 434
ASU_XP_LOGD_LOGDSN POF keyword 434
ASU_XP_LOGD_MGMTC POF keyword 434
ASU_XP_LOGD_PRIQTY POF keyword 434
ASU_XP_LOGD_SECQTY POF keyword 434
ASU_XP_LOGD_STORC POF keyword 434
ASU_XP_LOGD_UNIT POF keyword 434
ASU_XP_UIMSRVHOST POF keyword 434
ASU_XP_UIMSRVPORT POF keyword 435
ASU_XP_UIMSRVTIMEOUT POF keyword 435
ATS command 514
attach to a specified SSID 156
SSID
attaching 156
ATTACH OUTPUT descriptor variable
ATTACH symbolic variable 387
ZSYSID symbolic variable 398
ATTACH symbolic variable 387
attaching a CATALOG MANAGER session
from a connection server list 158
to a specified SSID 156
ATTR table 152
ATTR_VAL table 152
attribute
definition 528
ATW command 514
AU command 514
AU object type 36
AUC AEXIN keyword 377
AUDIT command 502
AUDIT installation option 404
audit logs 350
administrative functions 350
AUDIT_LOG table 152
audited events 350
AUDPOL installtion option 411
AUTHID
definition 528
authorization
required for object creation 239
required for SEARCH 178
restricting access through plans 64
setting 67
authorization ID
definition 528
authorization object type 36
authorizations, verifying 326
AUTO command 298
AUTOALE installation option 411
AUTORUN installation option 411
AUTOWIN installation option 411
Auxiliary and base tables object types 39
auxiliary index
definition 528
auxiliary list
definition 529
auxiliary table
definition 529
auxiliary table object type 36, 178
auxiliary table space
definition 529
AUXRELS installation option 411

B
B command 216
B line command 523
base table
definition 529
base table space
definition 529
baseline
definition 529
baseline name template
definition 529
baseline profile
definition 530
BATCH
CATALOG MANAGER lists 205
HDDL output data set 201
use with DDL, DESCRIBE 201
BATCH command 201, 502
batch component
definition 530
BDSN installation option 404
BIND command 66, 506
BIND DSN command 201
BINDCOPY command 506
BINDDEPLOY command 506
BINDFAIL AEXIN keyword 377
BINDFAIL POF keyword 435
block QQ command 374
BLRP data set 109
BLRP_DATACLASS POF keyword 435
BLRP_DATACLASS_ALT POF keyword 435
BLRP_EXPDT POF keyword 435
BLRP_MGMTCLASS POF keyword 435
BLRP_MGMTCLASS_ALT POF keyword 435
BLRP_PREFIX POF keyword 436
BLRP_PRIQTY POF keyword 436
BOPTS installation option 405
BP command 514
BP object type 36
BPLAN installation option 405
BR command 140, 216, 514
BROWSE command 216, 502
browsing logs 347
buffer pool object type 36

C
C line command 523
C2 command 514
C2 object type 37
CA command 514
CA object type 36
cabinet copy object type 39
CAN
   definition 531
CANCEL
   definition 531
CANCEL command 73, 502, 520
CANCEL ISPF command 43
carriage return, hexadecimal format 42
CASCADE BATCH command 201
CASCADE command 502
CAT_LOAD POF keyword 439
catalog access
   -l indicator (indirect) 46
   -R indicator (direct) 46
   -S indicator (server connection) 46
   switching between direct and indirect 158
catalog accessing
   setting 29
catalog baseline
   definition 531
catalog indirection 24, 158
   definition 531
CATALOG MANAGER
   commands, list of 501
   installation option descriptions 404
   installation options, list of 401
   plans, list of 64
   product code 64
   switches 90
CATALOG MANAGER for DB2
   definition 531
CATALOG MANAGER tables 152
catalog statistics 195
CATALOGHELP command 503
CATAUDIT AEXIN keyword 378
CATDOPT AEXIN keyword 378
CATOP installation option 405
CATRECOV installation option 379
CATRECOVER AEXIN keyword 379
CATSTATS command 195, 513
CATUTIL AEXIN keyword 379
CD command 514
CD object type 36, 178
CDL
   definition 531
CHANGE command 520
Change Definition Language CDL
   definition 531
CHANGE MANAGER for DB2
   definition 532
change rule
   definition 532
Character Field Max Width option field 79
characters
   nonprintable 42
   nonviewable 42
check 2 object type 37
CHECK command 276, 510
check constraint object type 178
check dependent object type 36, 178
CHECK INDEX command 276, 510
CHECK IX command 276, 510
check object type 36
CHECK+_LOAD POF keyword 440
CHECKDE installation option 411
CHECKDOPT POF keyword 440
CHECKOPT AEXIN keyword 379
checkpoint
   definition 532
CHECKS installation option 411
CHECKS2 installation option 411
CHGMAN_LOAD POF keyword 440
child
   definition 532
CHKD command 276, 510
CHKI command 276, 510
CI command 514
CI object type 36
CK command 514
CK object type 36, 178
CL command 514
CL object type 36
CLEANUP_RC POF keyword 440
cleanup, data set 100
CLIPBOARD command 312, 503
CLIST
  BMCADMF2 58, 223
  BMCDB2 70, 158, 223
definition 532
  sample 147
  variables 340
  writing user commands 146
CLIST command 503
CMD command 503
CMD_DISABLE command equivalent 142
CMPILOT for DB2
definition 532
CMPILOT worklist
definition 532
CNTL_DATACLASS POF keyword 440
CNTL_EXPDT POF keyword 440
CNTL_MGMTCLASS 440
CNTL_PREFIX POF keyword 441
CNTL_PRIQTY POF keyword 441
CNTL RETPD POF keyword 441
CNTL SECQTY POF keyword 441
CNTL_STORCLASS POF keyword 441
CNTL_UNIT POF keyword 441
CNTLMOUT_DSN POF keyword 441
CNTLMSCH_DSN POF keyword 441
CO command 514
CO object type 36, 176, 178
COLAUTH installation option 411
COLDISH installation option 411
COLDISS installation option 411
COLDIST installation option 411
collating sequence 193
collection object type 36
COLSTAT installation option 411
COLSTATS command 513
column authorization object type 36
column label object type 36
column object type 36, 178
COLUMNNH installation option 411
COLMNS installation option 411
combined lists
  command format 176
  excluding objects 177
  generating 176
  valid source objects 176
command
definition 533
COMMAND (CMD) command 52
command name (exccomnd) 147
command program, writing 147
command prompts 295
command recognition character 33
command reference 52
command syntax 295
commands
  invoking BMC Software utilities 275
    -BMCU (BMCU Execute a BMC Utility) 495
    -DBUG (Debug) 495
    -SQLM (SQL Statement) 499
  abbreviating 54
  adding 139
  CATALOG MANAGER, list of 501
  changing functionality 140
  command-line, list of 519
  data browsing 216
  data browsing, list of 519
  data editing 223
  data editing, list of 519
  disabling 139
  editing the commands table 140
  erasing leftover characters 55
  executing 52
  Fast Path Navigation 58
  including multiple objects 55
  invoking IBM DB2 utilities 276
  ISPF 43
  issuing from Cmd column 48, 55
  issuing from command line 32
  issuing similar 54
  line, list of 523
  list, list of 513
  listing 52
  omitting parameters 54
  parameters 54
  replacing format 140
  shortcuts 55
  typing 55
  user-written 141
  utility list, list of 512
  utility, list of 508
  valid for object list 53
  Wait-for-Enter (WFE) 55
COMANDS command 503
commands table
$ACTCMD macro 143
ACTCOMND member 138
adding user-written commands 141
log option (excclog) 147
modifying 139
retaining from previous release 146
REXX entry 143
syntax and parameters 143
tailoring 329
COMMENT command 506
commit
definition 533
common area, viewing 62
Common Explain component
integrating with CATALOG MANAGER 367
Compare
definition 533
Compare1
definition 533
Compare2
definition 533
compiling and linking user commands table 139
compiling SLIBs 363
compartment
definition 533
Confirm SQL panel
for extended SQL processing 212
setting default values 86
Conform SQL panel
working with SQL 207
CONNECT (CON) command
attaching to SSID 156
connecting to specified SSID 162, 163, 168
connect and attach functions, defined 155
CONNECT BATCH command 201
CONNECT command 170, 506
CONNECT RESET command 159
connecting to a specified SSID 162
connecting to an SSID
using a saved connection 166
Connections Table 168
CONSTDDE installation option 411
constraint
definition 533
constraint dependent object type 37
CONTAB command 168
CONTRL installation option 411
Control Accelerator Tables screen 502
control characters, using in object names 42
COPY command 276, 510, 520
COPY INDEX command 276, 510
COPY installation option 411
COPY IX command 276, 510
COPY+_LOAD POF keyword 441
COPYAUTHS command 304, 313, 503
COPYDOPT POF keyword 442
copying data
Copy Table Rows option 223
data compatibility 230
overview 230
using COPY command 230
using Copy Table Rows option 233
copying user privileges by privilege type 313
COPYOPT AEXIN keyword 379
COPYTOCOPY command 276, 510
COPYTOCOPY INDEX command 276, 510
COPYTOCOPY IX command 276, 510
COUNT command 195, 506
CP command 514
CP object type 37
CP object type, search qualifier 42
CPYEXP_DATACLASS POF keyword 442
CPYEXP_EXPDT POF keyword 442
CPYEXP_MGMTCLASS POF keyword 442
CPYEXP_PREFIX POF keyword 442
CPYEXP_RETPD POF keyword 442
CPYEXP_STORCLASS POF keyword 442
CPYEXP_SUPPRESS_SUFF POF keyword 442
CPYEXP_UNIT POF keyword 443
CR symbolic variable 387
CREATE command 506
creating an object hierarchy 255
creating and editing referential and unique
constraints 245
creating objects, using DDL commands 255
creating tables
copying table design 245
creating and editing constraints 245
defining columns 245
using an existing object 245
creating utility profile data sets 279
Cross-System Image Manager XIM
definition 534
CRS installation option 405
CRS option with session profiles 340
CRS_VAL table 152
CUP installation option 405
CUSTOMIZE (CU) command 330, 331, 338
CUSTOMIZE command 503
customizing CATALOG MANAGER
authorization 330
creating a session profile 335
CRS option 340
editing a tailored commands table 339
initial list filters 337
Primary Menu 331, 338
types of customization 329
customizing object list display 191
CUT command 312
CX command 514
CX object type 37, 178
CXA command 514
CXA object type 37
CXATTR installation option 411
CXAUTH installation option 412
CXT command 514
CXT object type 37
data editing commands, list of 519
data locking 226
Data Manipulation Language DML
definition 534
data set
map 100
data set information
obtaining on a remote SSID 164
data set object type 37
data set sizing 355
definition 534
data sets
cleanup 100
DATAWORK 107
deleting automatically 100
discard 100
error 100
LOGSORT 107
naming conventions 84
options 84
permanent work 109
prefixes 109
punch 100
resolving names 109
sample VSAM object 100
setting default processing options 109
setting options 84
sizing 100, 356
SORTOnnn 100
SORTOUT 100, 109
SORTPnnn 100
SORTWORK 100, 107
SUTnnn 100
SYSCOPIY 109
SYSDISC 109
SYSERR 109
SYSMAP 109
SYSPUNCH 109
SYSREC 109
SYSUT 109
SYSUTnnn 100
unload 100
work 100
WRKnnn 100
data structure
definition 535
data type object type 37, 178
DATA_PACKER_LOAD POF keyword 443
DATABAS installation option 412
Database Administration for DB2 26
database administrator DBA
definition 535
database object type 35, 178
database request module DBRM
definition 535
DATASETSIZING B keyword 101
DATASETSIZING C keyword 101
DATASETSIZING O keyword 101
DATASETSIZING POF keyword 443
DATATYP installation option 412
DATAWK_NBR POF keyword 444
DATAWK_UNIT POF keyword 444
DATAWORK data set 107
DATE OUTPUT descriptor variable
  DATE symbolic variable 387
  DT symbolic variable 388
  JYMD symbolic variable 391
  YMD symbolic variable 398
DATE symbolic variable 387
DATE TEMPLATE descriptor variable
  DATE symbolic variable 387
  JYMD symbolic variable 391
  YMD symbolic variable 398
DATEJ symbolic variable 387
DAY OUTPUT descriptor variable 387
DAY symbolic variable 387
DAY TEMPLATE descriptor variable 387
DB command 515
DB object type 35, 174, 176, 178
DB OUTPUT descriptor variable
  CR symbolic variable 387
  DB symbolic variable 388
  DBNAME symbolic variable 388
  IXCR symbolic variable 389
  TBCR symbolic variable 395
  TBCRE symbolic variable 395
  VCAT symbolic variable 397
DB symbolic variable 388
DB TEMPLATE descriptor variable
  CR symbolic variable 387
  DB symbolic variable 388
  DBNAME symbolic variable 388
  IXCR symbolic variable 389
  TBCR symbolic variable 395
  TBCRE symbolic variable 395
  VCAT symbolic variable 397
DB..IS TEMPLATE descriptor variable 389
DB..SN TEMPLATE descriptor variable 392, 393
DB..TS TEMPLATE descriptor variable 395
DB.TS OUTPUT descriptor variable
  IXNODE symbolic variable 389
  OBNAME symbolic variable 392
  OBNODE symbolic variable 393
  TBNODE symbolic variable 395
DB2
  initialization parameters, viewing 61
  plan name (excsplan) 147
  security 64
  special registers, viewing 62
  subsystem attached to (excsid) 147
DB2 accelerator commands 502
DB2 Attach
  attach and connect, defined 155
  Call Attach Facility (CAF) 155
  default attachment 159
  from a connection server list 158
  requirements 155
  restoring 159
DB2 catalog
definition 535
DB2 command
  command profiles 301
  definition 535
  PROFILE command 301
  specifying command syntax 299
DB2 commands
  issuing 295
  issuing on a remote SSID 165
  using command prompts 295
  using model commands 298
  using the PROFILE command 295
DB2 Connect
  change access panels 168
  Distributed Data Facility (DDF) 160
  requirements 159
  saved connections 166
  specified SSID 162
  using saved connection 166
DB2 DSN6 macros 61
DB2 Utilities
  DSN1COPY 496
  DB2-identifiers 170
DB2COMMAND command 507
DB2EXIT POF keyword 444
DB2LOAD POF keyword 444
DB2STMSGS AEXIN keyword 380
DB2V2 symbolic variable 388
DB2V3 symbolic variable 388
DBA
definition 535
DBAUTH installation option 412
DBCS
    definition 535
DBCS installation option 405
DBNAME symbolic variable 388
DBRM installation option 412
DBRM object type 36
DBRMs, explaining with Common Explain 369
DCL BATCH command 201
DCL command 317, 503
DCLGEN command 507
DD statement
    definition 535
DDD symbolic variable 388
DDL
    definition 535
DDL baseline
    definition 536
DDL BATCH command 201
DDL BATCH processing 201
DDL command 255, 503
ddbname
    definition 536
DDNAME symbolic variable 388
DDOPT symbolic variable 388
DDSEQ symbolic variable 388
debug worklist command 495
debugging, setting default processing options 117
DEBUGUNLD AEXIN keyword 380
Decimal Point option field 149
DEF_GDG_BASE POF keyword 444
DEF_GDG_LIMIT POF keyword 444
DEF_GDG_NOSCR POF keyword 444
DEF_GDG2_LIMIT POF keyword 445
default options module
    definition 536
Default SQUID option field 76
default value
    definition 536
defaults, setting product options
data sets 109
debugging 117
generation data groups 115
JCL jobcard 96
JCL static data sets 100
JCL STEPLIBs 98
LISTDEF data set 126
product options file 127
SORTWORK data sets 107
tapes 104
TEMPLATE data set 126
utility module names 126
defaults, setting user options
    basic options 76
data sets 109
debugging 115
general 67, 73
general options 79
generation data groups 115
non-worklist JCL 122
object use options 81
online reorgs 120
options data set 74
SHRLEVEL CHANGE 120
user variables 127
defining an options data set 74
defining table columns 245
DELETE (DEL) command 180
DELETE command 507
deleting
    data sets automatically 100
delimited identifier
    definition 536
delimited identifiers 79
DEPEND installation option 412
dependencies
    definition 536
dependency object type 37, 178
dependent
    definition 536
dependent object
    definition 536
DES BATCH command 201
DES command 198, 503
DESCRIBE BATCH command 201
DESCRIBE BATCH processing 201
DESCRIBE command 197, 199, 367, 370, 503
DESCRIBE options 93
DEST command 513
DESTATISTICS (DEST) command 199
DESTATISTICS BATCH command 201
DESTATISTICS command 513
destination  definition 536
development aids 147
device type  definition 537
DIAGMSGCLASS POF keyword 445
DIS command 507
disabling commands 139
DISCDATACLASS POF keyword 445
DISCDATACLASS_ALT POF keyword 445
DISCEXPDT POF keyword 445
DISCMGMTCLASS POF keyword 445
DISCMGMTCLASS_ALT POF keyword 445
DISC_PREFIX POF keyword 445
DISC_PRIQTY POF keyword 445
DISC_RETPD POF keyword 446
DISC_SECQTY POF keyword 446
DISC_STORCLASS POF keyword 446
DISC_STORCLASS_ALT POF keyword 446
DISC_THRESH POF keyword 446
DISC_UNIT POF keyword 446
DISC_UNIT_ALT POF keyword 446
discard data set 100
DISDATABASE command 507
DISALLOW_POPUP POF keyword 446
DISAUTO_TAB POF keyword 447
DISLOCATION POF keyword 447
DIS OMIT_CHAR POF keyword 447
DIS STATS POF keyword 447
DISVARDBG POF keyword 448
DISPLAY command 507
DISPLAYDB2command 201
displaying truncated utility profile ID 281
DISTHREAD command 507
DISTSTATS command 513
DISUTIL command 510
DLG table 152
DLG_ATTR table 152
DM command 515
DM object type 36
documentation  
field-level Help 44
documentation information 15
DOPTS  
definition 537
DOPTS command 67, 503
double-byte character set DBCS  
definition 537
DOWN command 520
DOWNISPFcommand 43
DP command 515
DP object type 37, 178
DPT installation option 406
DRO installation option 406
*DROP marker 264
DROP and DROPRECOVERY functions  
batch processing 268
excluding indexes from recovery 268
objects excluded from recovery 268
recovering multiple indexes 268
table spaces 264, 268
DROP command 264, 507
DROPIS switch 264
DROP NOT DONE message 264
Drop Recovery Log  
actions recorded 351
recording simulated DROP 263
dropping pending changes 261
dropping table spaces 261
DROPR command 268
DROPR NOIC POF keyword 382, 448
DROPRECOVERY command 503
DS command 515
DS object type 37
DSN1COMP command 276, 510
DSN1COPY command 276, 510
DSN1COPY utility 268, 496
DSNCHECK44 POF keyword 448
DSNEXIT library 61
DSNHDECPlparameter 61
DSNTIADPLANPOFkeyword 448
DSNUM OUTPUT descriptor variable 388
DSNUM symbolic variable 388
DSNWZP 507
DSNWZP stored procedure 160, 507
DSNZPARAM command 61
DSNZPARM parameter 61
DT command 515
DT object type 37, 178
DT symbolic variable 388
DTTEMPLATE descriptor variable 388
duplicate  
definition 537
dynamic allocation  
data set prefixes 109
dynamic SQL 178
DYNWORKUNIT AEXIN keyword 380
E

E command 223
EBCDIC collating sequence 193
ED command 140, 223, 292, 512, 515
EDIT command 223, 503, 520
edit procedure
  definition 537
editing the commands table 140
EDITOR_USERS table 152
EDITPROC
  definition 537
electronic documentation 15
EN command 515
EN object type 37, 178
END
  definition 537
END command 298
END command, creating and editing tables 520
END ISPF command 43
enquiry character, hexadecimal format 42
ENTER ISPF command 43
Enter key
  definition 538
ENV AEXIN keyword 380
ENVIRON installation option 412
ENVIRONMENT (ENVI) command 61
ENVIRONMENT command 503
Environment object type 178
environment variables object type 37
EPLAN installation option 406
ERR_DATACLASS POF keyword 448
ERR_DATACLASS_ALT POF keyword 448
ERR_EXPDT POF keyword 449
ERR_MGMTCLASS POF keyword 449
ERR_MGMTCLASS_ALT POF keyword 449
ERR_PREFIX POF keyword 449
ERR_PRIQTY POF keyword 449
ERR_RETPD POF keyword 449
ERR_SECQTY POF keyword 449
ERR_STORCLASS POF keyword 449
ERR_STORCLASS_ALT POF keyword 449
ERR_THRESHOLD POF keyword 450
ERR_UNIT POF keyword 450
ERR_UNIT_ALT POF keyword 450
error data set 100
ESC installation option 406
EVENTS AEXIN keyword 380
examples
  BMCU command 495
  DSN1 command 496
  SQLM command 499
excclog parameter 147
EXCCLOG parameter 148
exccnum parameter 147
EXCCNUM parameter 148
exccobjc parameter 147, 149
EXCCOBJC parameter 148
exccobjl parameter 147
EXCCOBJL parameter 148
exccomnd parameter 147
EXCOMND parameter 148
EXCSRC parameter 147
EXCCSRC parameter 148
EXCEPTIONS command 510
EXCESID parameter 148
EXCHANGE command 507
excluding objects from processing 55, 268
exccplan parameter 147
EXCLN parameter 148
excrcc parameter 147
EXCRC parameter 148
excsesid parameter 147
excssid parameter 147
EXCSSID parameter 148
exctsoid parameter 147
EXCTSOIDEX 148
EXEC command 276, 510
EXEC_LOAD POF keyword 450
Execution
  definition 538
execution, worklist
    cleanup job step for data sets 100
    deleting data sets automatically 100
exit routine
  definition 538
EXPLAIN command 503
extended SQL processing 212
external hexadecimal format 42

F

F command 503
F1 (HELP) 43
F10 (LEFT) 43, 522
F11 (RIGHT) 44, 523
F2 (SPLIT) 44
F3 (END) 43, 520
F4 (ZOOM) 44, 244, 523
F5 (RFIND) 522
F7 (UP) 44
F8 (DOWN) 43
F9 (SWAP) 44
Fast Path Navigation 223
FCMD symbolic variable 388
FCPY_DATACLASS POF keyword 450
FCPY_EXPDT POF keyword 450
FCPY_MGMTCLASS POF keyword 450
FCPY_PREFIX POF keyword 451
FCPY_PRIQTY POF keyword 451
FCPY_RETSPD POF keyword 451
FCPY_STORCLASS POF keyword 451
FCPY_SUPPRESS_SUFF POF keyword 109, 451
FCPY_UNIT POF keyword 451
FCRS table 152
FEW command 298
field procedure
definition 538
FIELDPROC
definition 538
FIELDS installation option 412
file tailoring 362
FILT_DATACLASS POF keyword 452
FILT_EXPDT POF keyword 452
FILT_MGMTCLASS POF keyword 452
FILT_PREFIX POF keyword 452
FILT_PRIQTY POF keyword 452
FILT_RETPD POF keyword 452
FILT_SECQTY POF keyword 452
FILT_STORCLASS POF keyword 452
FILT_UNIT POF keyword 452
FC command 367, 370, 504
GLOBAL variables object type 39
GMAP table 152
GMAP installation option 406
GMAP installation option 406
GMAP table 152
GDG (generation data group)
setting default processing options 115
specifying 109
symbolic variable 109
using for data sets 109
GDG symbolic variable 109, 388
GDG_MODEL POF keyword 453
general options, setting 79
generating batch JCL 205
generating JCL by batch processing 201
generation data group. See GDG
generation data group GDG
definition 539
GET command 367, 370, 504
global variables object type 39
GROUP ID
definition 539
GRPNM symbolic variable 388
H
hash value
definition 539
HASHFAIL AEXIN keyword 380
HASHFAIL POF keyword 453
HASHWARNRC AEXIN keyword 380
HASHWARNRC POF keyword 453
HC command 84, 504
HDAL installation option 406
HDDL
  AUTH switch 255
  batch processing 201
  including GRANT 255
HDDL BATCH command 201
HDDL command 255, 504
HDDL CONCAT command 255
HDDL output data set 201
HDIX installation option 406
HDPL installation option 406
HDSY installation option 407
HDTB installation option 407
HDTR installation option 407
HDTIS installation option 407
HDVW installation option 407
Help
  online 15
HELP ISPF command 43
hexadecimal format, null character 42
HGRANT BATCH command 201
HGRANT command 255, 504
Hierarchical GRANT (HGRANT) command 308
HISTORY command 513
HLQ
  definition 539
  HLQ.UBMCCNTL members
    product options file 415
  HM symbolic variable 388
  HMS symbolic variable 388
  HO symbolic variable 389
  HO TEMPLATE descriptor variable 389
  HO.MI TEMPLATE descriptor variable
    HM symbolic variable 388
    TIME4 symbolic variable 395
host variables
  in SQL model statements 209
  long values 185
  testing 212
  used in SEARCH 185
  using quotation marks 185
  valid values in DML statements 185
HOUR OUTPUT descriptor variable 389
HOUR symbolic variable 389
HOUR.MINUTE OUTPUT descriptor variable
  HM symbolic variable 388
  TIME4 symbolic variable 395
HPLAN installation option 407
I
  I line command 523
  IBM utilities
    DSN1COPY 496
  IC command 515
  IC object type 37, 176
  IC symbolic variable 389
  IC TEMPLATE descriptor variable
    IC symbolic variable 389
    ICTYPE symbolic variable 389
    JOBTYP symbolic variable 390
    OBJT symbolic variable 392
    OBJTYP symbolic variable 392
    RTYPE symbolic variable 393
    RUNTYP symbolic variable 394
    TYPE symbolic variable 396
  ICCOL installation option 407
  ICSYC installation option 407
  ICTYPE OUTPUT descriptor variable 389
  ICTYPE symbolic variable 389
  ICTYPE TEMPLATE descriptor variable 389
  IDAA 25
  identity column object type 38
  identity columns 245
  IEFUSI exit 66
  IL command 515
  IL object type 37, 178
  IM (data editing) command 522
  IM command 515
  IM object type 37, 176
  image copy
    definition 540
    image copy object type 37
    import
      definition 540
    Import
      definition 540
    IMPORT command 504
    IN command 515
    IN object type 37, 178
    inbound migrate profile
      definition 540
    INCLUDE_SYSPRIN2 POF keyword 453
including SEARCH in BATCH jobs 187
incremental DDL
definition 540
index and table space partition object type 38
index cleanup 37
index mixed object type 37
index object type 35, 178
index space object type 37
index space partition object type 37, 42
index space statistics object type 37
INDEXES installation option 412
indexes, recovering multiple 268
INDEXH installation option 412
INDEXPA installation option 412
INDEXPH installation option 412
INDEXSH installation option 412
INDEXST installation option 412
INDEXSTATS command 513
initial list filters 329
creating 337
retrieving 339
initial POF, refreshing 131
input streams 68
INSERT command 507
installation option 413
installation options
cATALOG MANAGER 401
installation options module
definition 540
installation requirements 29
internal format 42
internal table
definition 540
INVALID COMMAND message 374
IOALOAD1 POF keyword 454
IOALOAD2 POF keyword 454
IP addresses for a given location (DDF) object type 178
IP command 515
IP list object type 37
IP name object type 37
IP object type 37, 42
IPLIST installation option 412
IPNAMES installation option 412
IS command 515
IS object type 37
IS symbolic variable 389
IS TEMPLATE descriptor variable 389
ISPF
AJXPODAT macro 129
commands 43
commands in CATALOG MANAGER 245
file tailoring 362
file tailoring for JCL generation 355
help 43
sharepool variables 146
variables 149
variables in data set names 84
ISPF skeletons
definition 541
ISPFvariable 149
ISS command 515
ISS object type 37
ITERATIONMODE AEXIN keyword 381
IX (data editing) command 522
IX command 515
IX object type 35, 176, 178
IX symbolic variable 389
IXC command 515
IXC object type 37
IXCR symbolic variable 389
IXNAME symbolic variable 389
IXNODE symbolic variable 389
IXSPACE symbolic variable 389
j
jar contents object type 37
jar object type 37
JARCONT installation option 412
JAROBJT installation option 412
Java option object type 37
Java path object type 37, 178
JAVAPTH installation option 412
JAVOPTS installation option 412
JB command 515
JB object type 37
JC command 515
JC object type 37
JCL
definition 541
JCL DSN
definition 541
JCL Generation
data set sizing 356
ISPF file tailoring 355
product options 70
RUNSTATS utility 356
setting options 83
VSAM object sampling 356
JCL Generation JCLGEN
definition 541
JCL job card, setting default processing options 96
JCL job statement, defaults 96
JCL static data sets, setting default processing options 100
JCL STEPLIBs, setting default processing options 98
JCL variable display
definition 541
JCLCLEANUP POF keyword 454
JCLGEN
definition 541
JCLLIB POF keyword 454
JD symbolic variable 389
JDATE OUTPUT descriptor variable
DATEJ symbolic variable 387
JDATE symbolic variable 389
JU symbolic variable 391
JUL4Y symbolic variable 391
JULIAN symbolic variable 391
YYDDDD symbolic variable 398
YYYYYDDD symbolic variable 398
JDATE symbolic variable 389
JDATE TEMPLATE descriptor variable
DATEJ symbolic variable 387
JDATE symbolic variable 389
JU symbolic variable 391
JUL4Y symbolic variable 391
JULIAN symbolic variable 391
YYDDDD symbolic variable 398
YYYYYDDD symbolic variable 398
JDAY OUTPUT descriptor variable
DDD symbolic variable 388
JDAY symbolic variable 389
JJULD symbolic variable 389
JDAY symbolic variable 389
JDAY TEMPLATE descriptor variable
DDD symbolic variable 388
JDAY symbolic variable 389
JJULD symbolic variable 389
JDDN symbolic variable 389
JDSN installation option 407
JES3 POF keyword 454
JHMS symbolic variable 389
JJULD symbolic variable 389
job
definition 541
JOB_INCLUDE_MEMBER POF keyword 454
JOBCARD1 POF keyword 455
JOBCARD2 POF keyword 455
JOBCARD3 POF keyword 455
JOBCARD4 POF keyword 455
JOBCARD5 POF keyword 455
JOBCHAR symbolic variable 390
JOBNAME OUTPUT descriptor variable
JOBCHAR symbolic variable 390
JOBNAME symbolic variable 390
JPCOD symbolic variable 390
MEMBER symbolic variable 392
MEMBR symbolic variable 392
PGMR symbolic variable 393
WKOWN symbolic variable 397
WKOWNER symbolic variable 397
JOBNAME symbolic variable 390
JOBNAME TEMPLATE descriptor variable
ATTACH symbolic variable 387
DDOPT symbolic variable 388
JDDN symbolic variable 391
JOBCHAR symbolic variable 390
JOBNAME symbolic variable 390
JPCOD symbolic variable 390
MEMBER symbolic variable 392
MEMBR symbolic variable 392
PGMR symbolic variable 393
WKOWN symbolic variable 397
WKOWNER symbolic variable 397
ZACCTNUM symbolic variable 398
JOBTYP symbolic variable 390
JOIN command 504
JP command 515
JP object type 37
JPCOD symbolic variable 390
JQID symbolic variable 390
JS1 symbolic variable 390
JS2 symbolic variable 390
JS4 symbolic variable 390
JSSID symbolic variable 391
JT command 515
JT object type 37, 178
JU symbolic variable 391
JUL4Y symbolic variable 391
JULIAN symbolic variable 391
JYMD symbolic variable 391
K

KC (data editing) command 522
KC command 515
KC object type 37
KCOLUSE installation option 412
Key -targets of extended indexes object type 178
key column object type 37
key column user object type 38
key-target distribution history object type 39
key-target distribution object type 37
key-target distribution statistics history object type 39
key-target history object type 38
key-target object type 37
KEY -targets of extended indexes object type 178
key column user object type 38
KEY TDST installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
KEY TGDS installation option 412
K PLAN installation option 407
KT command 516
KT object type 37, 178
KTD command 516
KTD object type 37
KTDH command 516
KTDH object type 39
K TDS command 516
K TDS object type 39
KTH command 516
KTH object type 38
KTS command 516
KTS object type 38
KU command 516
KU object type 38

L

LABEL command 507
large object LOB column
definition 542
large object LOB table space
definition 542
LCAT command 519
LDSN installation option 408
LDSNUM symbolic variable 391
LEDIT command 504
LEFT command 522
LEFT ISP F command 43
LI MX command 516
LI symbolic variable 391
LI TEMPLATE descriptor variable 391
limit key object type 38
LINES AEXIN keyword 381
LINK library
definition 542
LIST action, default 34, 47
LIST command 516
list commands, list of 513
list object type(exccobjl) 147
LIST symbolic variable 391
LIST TEMPLATE descriptor variable 391
LISTDEF control statements 291
LISTDEF data set 126
LISTDEF symbolic variable 391
LISTDEF_DSN POF keyword 455
lists
  printing 201
combined with SQL 209
customizing displays 191
describing 196
mixed lists 173
using Quick-Search 187
using SEARCH 178
literal strings 42
LK command 516
LK object type 38
LL command 516
LL object type 38
LL_CLIB POF keyword 455
LL_CLIB2 POF keyword 455
LL_CLIB3 POF keyword 455
LL_CLIB4 POF keyword 455
LL_CLIB5 POF keyword 455
LL_LINK POF keyword 455
LL_LINK2 POF keyword 455
LL_LINK3 POF keyword 455
LL_LINK4 POF keyword 455
LL_LINK5 POF keyword 455
LL_MLIB POF keyword 455
LL_MLIB2 POF keyword 455
LL_MLIB3 POF keyword 455
LL_MLIB4 POF keyword 455
LL_MLIB5 POF keyword 455
LL_PLIB POF keyword 455

Index 573
accessing 345
administering audit logs 350
administrative functions 345
browsing 347
DDL Audit Log 350
Drop Recovery Log 351
enable session 143
purging 349
Session Log 346, 347
types maintained 345
LOGSORT data set 107
LOGWK_NBR POF keyword 457
LOGWK_UNIT POF keyword 457
long names, truncating 79
LPART OUTPUT descriptor variable 391
LPART symbolic variable 391
LPLAN installation option 408
LR symbolic variable 391
LR TEMPLATE descriptor variable 391
LS command 516
LS object type 38, 178
LU command 516
LU list object type 38
LU mode object type 38, 178
LU mode select object type 38, 178
LU name object type 38, 178
LU object type 38, 178
LULIST installation option 413
LUMODES installation option 413
LUNAMES installation option 413

M

M line command 523
macros
$ACTEXC 147
$ACTULOG 147
AJXPEDAT 129
MAINT command 349
MAINTAIN command 345, 349, 504
MANY command 298
map data set 100
MAP_DATACLASS POF keyword 457
MAP_DATACLASS_ALT POF keyword 457
MAP_EXPDT POF keyword 457
MAP_MGMTCLASS POF keyword 457
MAP_MGMTCLASS_ALT POF keyword 457
MAP_PREFCLASS POF keyword 457
MAP_PREFIX POF keyword 457
MAP_PRIQTY POF keyword 457
MAP_RETPD POF keyword 458
MAP_SECQTY POF keyword 458
MAP_STORCLASS POF keyword 458
MAP_STORCLASS_ALT POF keyword 458
MAP_THRESH POF keyword 458
MAP_UNIT POF keyword 458
MAP_UNIT_ALT POF keyword 458
mapping database 120
mapping table 120
materialized query table object type 38
Materialized query table object type 178
MAX installation option 408
MAX_CYL POF keyword 458
MAX_PRIQTY POF keyword 459
MAX_SECQTY POF keyword 459
MAX_UNITCNT POF keyword 459
MDDL command 255, 504
MEMBER symbolic variable 392
MEMBR symbolic variable 392
MEMLIMIT POF keyword 459
MEMLIMIT system parameter 66
menu
   definition 542
MERGECOPY command 276, 510
MERGECOPY worklist command 496
message files 132
message, invalid command 374
MI symbolic variable 392
MI TEMPLATE descriptor variable 392
migrate
   definition 543
migrate profile
   definition 543
migrate-type work ID
   definition 543
MINUTE OUTPUT descriptor variable 392
MINUTE symbolic variable 392
MINUTE TEMPLATE descriptor variable 392
mixed list
   definition 543
mixed object lists
   excluding objects 174
   generating 174
   valid source objects 174
mixed object type 38
MK command 516
MMDD symbolic variable 392
MO symbolic variable 392
MO TEMPLATE descriptor variable 392
model commands 295
models
   creating new objects 55
   creating tables 245
MODESEL installation option 413
MODIFY command 276, 510
MODIFY STATISTICS worklist command 497
MODIFYRECOVERY command 276, 510
MODIFYSTATISTICS command 511
MODIFYSTATS command 276
MONTH OUTPUT descriptor variable 392
MONTH symbolic variable 392
MONTH TEMPLATE descriptor variable 392
MONTH.DAY OUTPUT descriptor variable 392
MONTH.DAY TEMPLATE descriptor variable 392
MORE command 226, 522
MPLAN installation option 408
MQT command 516
MQT object type 38, 178
MSG table 152
MSSID symbolic variable 392
multitasking, using tape stacking 104
MX command 174, 516
MX object type 38

N

name fields in user-written commands 149
NAME parameter 149
name propagation
   definition 543
name template
   definition 543
native SQL procedure object type 38, 178
navigating CATALOG MANAGER 45
negative acknowledgement, hexadecimal format 42
NEWTASKID AEXIN keyword 381
NEWWORKID AEXIN keyword 381
No operation worklist command 497
NOAPFOK AEXIN keyword 381
NOFAILNOIMAGECPY AEXIN keyword 382
NOLOADCOMP AEXIN keyword 382
non-worklist JCL 122
NONE command 298
nonprintable characters 42
nonviewable characters 42
NOSQLCOMP AEXIN keyword 382
NOWORKID AEXIN keyword 381
NOWORKID AEXIN keyword 381
NOTIFYUNLD AEXIN keyword 382
NOWKIDREPLACE AEXIN keyword 382

Index 575
NP command 516
NP object type 38, 178
null
definition 543
null characters 42
number of objects option(exccnum) 147

OB command 516
OB object type 38, 178
OBDS installation option 413
object
definition 544
object list
definition 544
object lists
customizing display 191
excluding objects 55
from different source object types 48
from multiple source objects 48
generating from Primary Menu 47
reordering columns 191
secondary lists 46
selecting objects 55
sorting by column 193
object role dependencies object type 178
object role dependency object type 38
Object role object type 178
object type in command text(exccobjc) 147
object types
generating lists 35
passing in user-written commands 149
OBJIROL installation option 413
OBJT symbolic variable 392
OBJTYP symbolic variable 392
OBNAM symbolic variable 392
OBNOD symbolic variable 393
online Help 15, 44
online reorgs 120
online schema changes object type 38
Online schema changes object type 178
OPT command (data editing and browsing) 522
OPT parameter 149
option
definition 544
options
changing 67
data editing and browsing 216
data set names 84
data sets 109
debugging and display 117
DESCRIBE 93
general 79
generation data group 115
installation 67
JCL Generation 83
JCL job cards 96
JCL static data sets 100
JCL STEPLIBs 98
LISTDEF data set 126
non-worklist JCL 122
online reorg 120
option switches 90
panel attributes 89
product options file 127
refreshing 68
setting 67
setting values 73
SHRLEVEL CHANGE 120
SORTWORK data sets 107
SQL command 86
SQL SELECT 88
tapes 104
TEMPLATE data set 126
used by products 72
user 68
user variables 127
utility module names 119
OPTIONS command 504
options data set, defining 74
ORDER command 504
ordinary identifiers 79
origin
definition 544
orphaned auxiliary index
definition 544
orphaned auxiliary table space
definition 544
ORTPARM_DSN POF keyword 459
OS command 517
OS object type 38
outbound migrate profile
definition 544
OUTPUT descriptor variables, list of 386
PART5 symbolic variable 393
partitioned data set PDS
definition 545
partitioned table space
definition 545
PASTE command 312
pattern
definition 545
PB symbolic variable 393
PB TEMPLATE descriptor variable 393
PCPY1_DATACLASS POF keyword 459
PCPY1_DATACLASS_ALT POF keyword 459
PCPY1_EXPDT POF keyword 459
PCPY1_MGMTCLASS POF keyword 459
PCPY1_MGMTCLASS_ALT POF keyword 459
PCPY1_PREFIX POF keyword 460
PCPY1_PRIQTY POF keyword 460
PCPY1_RETPD POF keyword 460
PCPY1_SECQTY POF keyword 460
PCPY1_STACK POF keyword 460
PCPY1_STORCLASS POF keyword 460
PCPY1_STORCLASS_ALT POF keyword 460
PCPY1_SUPPRESS_SUFF POF keyword 109, 460
PCPY1_THRESH POF keyword 460
PCPY1_UNIT POF keyword 461
PCPY1_UNIT_ALT POF keyword 461
PCPY2_DATACLASS POF keyword 461
PCPY2_DATACLASS_ALT POF keyword 461
PCPY2_EXPDT POF keyword 461
PCPY2_MGMTCLASS POF keyword 461
PCPY2_MGMTCLASS_ALT POF keyword 461
PCPY2_PREFIX POF keyword 461
PCPY2_PRIQTY POF keyword 461
PCPY2_RETPD POF keyword 462
PCPY2_SECQTY POF keyword 462
PCPY2_STACK POF keyword 462
PCPY2_STORCLASS POF keyword 462
PCPY2_STORCLASS_ALT POF keyword 462
PCPY2_SUPPRESS_SUFF POF keyword 109, 462
PCPY2_THRESH POF keyword 462
PCPY2_UNIT POF keyword 462
PDD command 517
PDISTSTATS command 513
PDS
definition 545
PDSN installation option 408
PEEK command 62
pending changes, dropping 261
percent character, hexadecimal format 42
permanent work data sets 109
*PERSIST option 76
PFSHOW ISPF command 43
PG command 517
PG object type 36, 176, 178
PGC command 517
PGMR symbolic variable 393
PI command 176, 517
PI object type 38
PK command 517
PK object type 38
PL command 517
PL object type 36, 174, 176, 178
plan authorization object type 38
plan object type 36, 178
plans
   CATALOG MANAGER 64
      explaining with Common Explain 369
      manipulating 66
      package use 176
PLP installation option 408
PM command 517
PM object type 38
POF
   AJXPODAT macro 129
      creating a user POF 127
      setting default processing options 127
      updating a user POF 129
POF (product options file)
   adding steps to JCL 135
   description 415
   generating reports 132
   initial 70
   initializing 70
   keyword descriptions 423
   keywords, list of 415
   overriding values in SLIBs 134
   POF Validation Report 132
   populating 70
   refreshing the initial POF 131
   reusing in a subsequent installation 133
   sample file 415
   user 70
   using multiple POFs 130
      Variables Initialized with Default report 132
POFDATE POF keyword 70, 463
POFDS installation option 70, 408, 415
pop-up window 33
PR command 517
PR object type 36, 178
PR parameter in ACTEMAIN CLIST 340
PRE_JOBSTEP_INCLUDE POF keyword 463
PREFIX symbolic variable 393
prefixes 109
PRI command 201
PRIBAC symbolic variable 393
PRIBAC TEMPLATE descriptor variable 393
primary key object type 38
PRINT CLOSE command 84
PRINT command 84, 201, 504
privilege
   definition 545
privileges
   copying by privilege type 313
   granting table privileges 304
   to create objects 239
privileges, reassigning 326
privileges, revoking 326
PRO command 505
PROC_BMCCHECK_NAME POF keyword 463
PROC_BMCHECK_STEP POF keyword 463
PROC_BMCCOPY_NAME POF keyword 463
PROC_BMCCOPY_STEP POF keyword 463
PROC_BMCCPRS_NAME POF keyword 463
PROC_BMCCPRS_STEP POF keyword 464
PROC_BMCMLOAD_NAME POF keyword 464
PROC_BMCMLOAD_STEP POF keyword 464
PROC_BMCRECOVER_NAME POF keyword 464
PROC_BMCRECOVER_STEP POF keyword 464
PROC_BMCREORG_NAME POF keyword 465
PROC_BMCREORG_STEP POF keyword 465
PROC_BMCSTATS_NAME POF keyword 465
PROC_BMCSTATS_STEP POF keyword 465
PROC_BMCSTOP_NAME POF keyword 465
PROC_BMCSTOP_STEP POF keyword 465
PROC_BMCTRIG_NAME POF keyword 466
PROC_BMCTRIG_STEP POF keyword 466
PROC_BMCUNLOAD_NAME POF keyword 466
PROC_BMCUNLOAD_STEP POF keyword 466
PROC_BMCUPS_NAME POF keyword 466
PROC_BMCUPS_STEP POF keyword 466
PROC_DSN1COPY_NAME POF keyword 467
PROC_DSN1COPY_STEP POF keyword 467
PROC_DSNUTILB_NAME POF keyword 467
PROC_DSNUTILB_STEP POF keyword 467
PROC_GEN_SET_VAR POF keyword 467
PROC_IDCAMSS_NAME POF keyword 468
PROC_IDCAMSS_STEP POF keyword 468
PROC_IEFBR14_NAME POF keyword 468
PROC_IEFBR14_STEP POF keyword 468
PROC_TSO_NAME POF keyword 468
PROC_TSO_STEP POF keyword 468
PROC_USE POF keyword 469
PROC_USER_DEF_STEP POF keyword 469
PROC_USER_DEFINED POF keyword 469
procedure object type 178
product Help 44
product options file. See POF
product options file.. See POF
profile
qualification
PROFILE command 295, 301, 330, 505
PROFILE command(session profiles) 333
PROFILE command(utility profiles) 291
PROFILE SAVE command 289
PROFILE SAVEAS command 281, 286, 287
PROFILE SET command 291
PROFILE.profileName ADDED message 331, 335, 337
PROFILE.profileName UPDATE message 338, 339
PROFILES command 505
protected baseline
PT command 517
PT object type 38, 176
PTF 61
publications, related 15
punch data set 100
PUNCH_DATACLASS 469
PUNCH_EXPDT POF keyword 469
PUNCH_MGMTCLASS POF keyword 469
PUNCH_MGMTCLASS_ALT POF keyword 470
PUNCH_PREFIX POF keyword 469
PUNCH_PRIQTY POF keyword 469
PUNCH_RETPD POF keyword 470
PUNCH_SECQTY POF keyword 470
PUNCH_STORCLASS POF keyword 470
PUNCH_UNIT POF keyword 470
PURGE COMPLETED message 349
purging logs 349
purging the session log 351

Q
QCONNECT command 507
QQ command 374
QRO command 517
QRP command 517
QRY command 517
QUAL parameter 149
Qualifier field
IP object type 42
object names 40
objects with two-part names 41
TP object type 42
wildcard characters 40
QUERY installation option 413
quick-search
using saved search variables 188
WHERE clause 188
quickname 166
QUIESCE command 276, 511
QUIESCE utility worklist command 498

R
R line command 523
RC parameter 149
RCCOL installation option 409
RCHANGE command 522
RCPY1_DATACLASS POF keyword 470
RCPY1_DATACLASS_ALT POF keyword 470
RCPY1_EXPDT POF keyword 470
RCPY1_MGMTCLASS POF keyword 470
RCPY1_MGMTCLASS_ALT POF keyword 470
RCPY1_PREFIX POF keyword 471
RCPY1_PRIQTY POF keyword 471
RCPY1_RETPD POF keyword 471
RCPY1_SECQTY POF keyword 471
RCPY1_STACK POF keyword 471
RCPY1_STORCLASS POF keyword 471
RCPY1_STORCLASS_ALT POF keyword 471
RCPY1_SUPPRESS_SUFF POF keyword 109, 471
RCPY1_THRESH POF keyword 471
RCPY1_UNIT POF keyword 472
RCPY1_UNIT_ALT POF keyword 472
RCPY2_DATACLASS POF keyword 472
RCPY2_DATACLASS_ALT POF keyword 472
RCPY2_EXPDT POF keyword 472
RCPY2_MGMTCLASS POF keyword 472
RCPY2_MGMTCLASS_ALT POF keyword 472
RCPY2_PREFIX= POF keyword 472
RCPY2_PRIQTY POF keyword 472
RCPY2_RETPD POF keyword 473
RCPY2_SECQTY POF keyword 473
RCPY2_STACK POF keyword 473
RCPY2_STORCLASS POF keyword 473
RCPY2_STORCLASS_ALT POF keyword 473
RCPY2_SUPPRESS_SUFF POF keyword 109, 473
<table>
<thead>
<tr>
<th>Command</th>
<th>Page Numbers</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCPY2_THRESH</td>
<td>473</td>
<td>POF keyword</td>
</tr>
<tr>
<td>RCPY2_UNIT</td>
<td>473</td>
<td>POF keyword</td>
</tr>
<tr>
<td>RCPY2_UNIT_ALT</td>
<td>473</td>
<td>POF keyword</td>
</tr>
<tr>
<td>RD command</td>
<td>517</td>
<td></td>
</tr>
<tr>
<td>RD object type</td>
<td>38, 178</td>
<td></td>
</tr>
<tr>
<td>RE command</td>
<td>517</td>
<td></td>
</tr>
<tr>
<td>RE object type</td>
<td>38, 176</td>
<td></td>
</tr>
<tr>
<td>real-time statistics.</td>
<td></td>
<td>See RTS</td>
</tr>
<tr>
<td>REBIND command</td>
<td>66, 508</td>
<td></td>
</tr>
<tr>
<td>REBIND DSN command</td>
<td>201</td>
<td></td>
</tr>
<tr>
<td>REBINDFAIL AEXIN</td>
<td>382</td>
<td>keyword</td>
</tr>
<tr>
<td>REBINDFAIL POF</td>
<td>474</td>
<td>keyword</td>
</tr>
<tr>
<td>REBINDRC AEXIN</td>
<td>383</td>
<td>keyword</td>
</tr>
<tr>
<td>REBINDRC POF</td>
<td>474</td>
<td>keyword</td>
</tr>
<tr>
<td>REBUILD command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>REBUILD INDEX command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>REBUILD IX command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>REC DATA text line</td>
<td>268</td>
<td></td>
</tr>
<tr>
<td>REC LRBA comment</td>
<td>268</td>
<td></td>
</tr>
<tr>
<td>receive-type work ID</td>
<td>546</td>
<td>definition</td>
</tr>
<tr>
<td>record layout</td>
<td>493</td>
<td></td>
</tr>
<tr>
<td>RECOVER command</td>
<td>268, 276, 511</td>
<td></td>
</tr>
<tr>
<td>RECOVER INDEX command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>RECOVER IX command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>REORG command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>REORG INDEX command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>REORG IX command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>REORG_MAPDB POF</td>
<td>474</td>
<td>keyword</td>
</tr>
<tr>
<td>REORG_MAPTAB POF</td>
<td>475</td>
<td>keyword</td>
</tr>
<tr>
<td>REORGDOPT POF</td>
<td>475</td>
<td>keyword</td>
</tr>
<tr>
<td>REORGOPT AEXIN</td>
<td>383</td>
<td>keyword</td>
</tr>
<tr>
<td>REPAIR utility worklist command</td>
<td>499</td>
<td></td>
</tr>
<tr>
<td>REPLACETASKID AEXIN</td>
<td>383</td>
<td>keyword</td>
</tr>
<tr>
<td>REPLACEWORKID AEXIN</td>
<td>383</td>
<td>keyword</td>
</tr>
<tr>
<td>REPORT command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>REPORT INDEX command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>REPORT IX command</td>
<td>276, 511</td>
<td></td>
</tr>
<tr>
<td>REPORT utility worklist command</td>
<td>498</td>
<td></td>
</tr>
<tr>
<td>reports, generating POF</td>
<td>132</td>
<td></td>
</tr>
<tr>
<td>REPT_DATACLASS POF</td>
<td>475</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_DATACLASS_ALT</td>
<td>475</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_EXPDT POF</td>
<td>476</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_MGMTCLASS POF</td>
<td>476</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_MGMTCLASS_ALT</td>
<td>476</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_PREFIX POF</td>
<td>476</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_PRIQTY POF</td>
<td>476</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_RETPD POF</td>
<td>476</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_SECQTY POF</td>
<td>476</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_STORCLASS POF</td>
<td>476</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_STORCLASS_ALT</td>
<td>476</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_THRESH POF</td>
<td>477</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_UNIT POF</td>
<td>477</td>
<td>keyword</td>
</tr>
<tr>
<td>REPT_UNIT_ALT POF</td>
<td>477</td>
<td>keyword</td>
</tr>
<tr>
<td>RESAUTH installation option</td>
<td>413</td>
<td></td>
</tr>
<tr>
<td>RESET command</td>
<td>505</td>
<td></td>
</tr>
<tr>
<td>RESET subcommand</td>
<td>159</td>
<td></td>
</tr>
<tr>
<td>RES/RES command</td>
<td>522</td>
<td></td>
</tr>
<tr>
<td>RESPONSES command</td>
<td>505</td>
<td></td>
</tr>
<tr>
<td>RESTART AEXIN</td>
<td>383</td>
<td>keyword</td>
</tr>
<tr>
<td>RESTARTPARAM AEXIN</td>
<td>383</td>
<td>keyword</td>
</tr>
<tr>
<td>retaining user commands from previous release</td>
<td>146</td>
<td></td>
</tr>
<tr>
<td>RETURN ISPF command</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>REVOKE command</td>
<td>318, 508</td>
<td></td>
</tr>
<tr>
<td>REXX commands table entry</td>
<td>143</td>
<td></td>
</tr>
<tr>
<td>RFIND command</td>
<td>522</td>
<td></td>
</tr>
<tr>
<td>RHLQ symbolic variable</td>
<td>393</td>
<td></td>
</tr>
<tr>
<td>RI command</td>
<td>517</td>
<td></td>
</tr>
<tr>
<td>RI object type</td>
<td>38</td>
<td></td>
</tr>
</tbody>
</table>

Remote SSID connecting 164, 165 connecting; DB2 commands 165
RENAME command 508
reordering object list columns 191
REORG command 276, 511
REORG INDEX command 276, 511
REORG IX command 276, 511
REORG_MAPDB POF keyword 474
REORG_MAPTAB POF keyword 475
REORG+_LOAD POF keyword 475
REORGDOPF POF keyword 475
REORGOPT AEXIN keyword 383
REPAIR utility worklist command 499
REPLACETASKID AEXIN keyword 383
REPLACEWORKID AEXIN keyword 383
REPORT command 276, 511
REPORT INDEX command 276, 511
REPORT IX command 276, 511
REPORT utility worklist command 498
reports, generating POF 132
REPT_DATACLASS POF keyword 475
REPT_DATACLASS_ALT POF keyword 475
REPT_EXPDT POF keyword 476
REPT_MGMTCLASS POF keyword 476
REPT_MGMTCLASS_ALT POF keyword 476
REPT_PREFIX POF keyword 476
REPT_PRIQTY POF keyword 476
REPT_RETPD POF keyword 476
REPT_SECQTY POF keyword 476
REPT_STORCLASS POF keyword 476
REPT_STORCLASS_ALT POF keyword 476
REPT_THRESH POF keyword 477
REPT_UNIT POF keyword 477
REPT_UNIT_ALT POF keyword 477
RESAUTH installation option 413
RESET command 505
RESET subcommand 159
RESET/RES command 522
RESPONSES command 505
RESTART AEXIN keyword 383
RESTARTPARAM AEXIN keyword 383
retaining user commands from previous release 146
RETURN ISPF command 44
REVOKE command 318, 508
REXX commands table entry 143
RFIND command 522
RHLQ symbolic variable 393
RI command 517
RI object type 38
RIGHT command 523
RIGHT ISPF command 44
RO command 517
RO object type 38, 178
ROLES installation option 413
roles object type 38
ROUTINA installation option 413
ROUTE installation option 413
routine object type 178
routine option object type 37
routine parameter object type 37
routine source object type 37
ROUTOPT installation option 413
ROUTSRC installation option 413
row permissions 38
ROWID SYSREC data set 109
ROWVIEW/ROW command 523
RP command 290, 512
RSEQ# symbolic variable 393
RTS (real-time statistics)
  index space 518
  table space 518
RTYPE symbolic variable 393
rules
  DSN1 command 496
RUNSTATS command 276, 511
RUNSTATS INDEX command 276, 511
RUNSTATS IX command 276, 511
RUNSTATS utility 356
RUNTIME_HLQ POF keyword 477
RUNTYP symbolic variable 394

S
S command 505
SAVE command 523
SBCS
  definition 547
SC command 517
SC object type 38
SC symbolic type 38
SC symbolic type 394
SC TEMPLATE descriptor variable 394
SCHEMAA installation option 414
scope
  definition 547
scope rule
  definition 547
Script table
  definition 547
SE command 517
SE object type 38, 178
SEARCH
  batch jobs 187
SEARCH command 178, 505
SEARCH function 178
  associate with profile 180
  associate with user ID 180
  authorization required 178
  complex subqueries 189
  generating lists 180
  JOINed tables 190
  operator (Oper) variables 180
  retrieving saved variables 180
  saving variables 180
  setting values 180
  using host variables 185
  valid source objects 178
  WHERE clauses 180, 185
SEARCH_VARS2 table 152
SEC OUTPUT descriptor variable 394
SEC symbolic variable 394
SECOND OUTPUT descriptor variable 394
SECOND symbolic variable 394
SECOND TEMPLATE descriptor variable 394
secondary lists, generating from object lists 48
security, setting authorizations 67
SEE command 62, 508
SELECT command 508
selection list
  definition 548
SEQ OUTPUT descriptor variable
  DDPOT symbolic variable 388
  DDSEQ symbolic variable 388
  JDDN symbolic variable 389
  RSEQ# symbolic variable 393
  SEQ symbolic variable 394
  SEQ# symbolic variable 394
  SQ symbolic variable 394
SEQ symbolic variable 394
SEQ TEMPLATE descriptor variable
  DDSEQ symbolic variable 388
  RSEQ# symbolic variable 393
  SEQ symbolic variable 394
  SEQ# symbolic variable 394
SEQ# symbolic variable 394
SEQAUTH installation option 414
SEQDEP installation option 414
SEQUENC installation option 414
sequence number
  definition 548
sequence object type 178
SESSION command 505
session logs
  actions captured 346
  browsing 347
  enabling 143
  purging 349
session profiles 329
  assigning to users 340
  authorization to create 330
CLIST 340
  creating initial list filters 337
CRS option 340
CUSTOMIZE command 331
  customizing commands table 335
  customizing Primary Menu 331
  deleting 337
  displaying descriptions 334
  displaying list 333
  editing customized commands table 339
  editing customized Primary Menu 338
PR parameter in ACTEMAIN CLIST 340
SET PROFILE command 330
TAILOR command 335
  types of customization 329
SESSION_LOG table 152
SET command 505
SG command 517
SG object type 35, 174, 176, 178
SHRLEVEL CHANGE 120
Simple Space Estimation (SSE) feature 44, 239
  single-byte character set SBCS
  definition 548
skeleton library compiler. See SLIB (skeleton library) compiler
skeleton library, overriding POF values 134
SLIB (skeleton library) compiler
  changing an ISPF skeleton 361
  compiling 361, 363
  ISPF file tailoring 362
  processing 363
  runtime report 364
  runtime report summary 364
  runtime unit 363
  testing changes 362
SLIB variables, list of 386
SMFPRM member 66
SMS (Storage Management Subsystem) 109
SN symbolic variable 394
SN TEMPLATE descriptor variable 394, 395
  solutions, BMC Software 26
SORT command 193, 505, 523
  sorting object lists by column 193
SORTOnnn data set 100
SORTOUT data set 100, 109
SORTPnnn data set 100
SORTWK_NBR POF keyword 477
SORTWK_PRIQTY POF keyword 477
SORTWK_SECQTY POF keyword 478
SORTWK_UNIT POF keyword 478
SORTWORK data set 107
  source
    definition 548
SPACE command 511
Space Estimation
  definition 548
SPBXPRINT AEXIN keyword 383
Specification
  definition 548
specify log option (OPT) 149
SPLAN installation option 409
SPLIT ISPF command 44
SPNAME symbolic variable 394
SQ symbolic variable 394
SQ TEMPLATE descriptor variable 394
SQL
  applying model statements 209
  copying external 209
  copying in SQL_Table 208
  creating new SQL from existing 209
  creating new SQL_Table entry 208
  definition 549
  escape character 79
  extended SQL processing 212
  generating 206
  SQL_Table 207
SQL command 505
SQL command options, setting 86
SQL Explorer (for DB2)
  ACTPSS CLIST 367
  commands to access SQL 367
  SQLX edit macro 374
SQL Explorer for DB2
  integrating with CATALOG MANAGER 367
SQL SELECT options, setting 88
SQL_Table
copying SQL 208
creating new entry 208
defined 207
displaying 207
explaining a statement with Common Explain 373
SQL_TABLE table 152
SQLEXP_LOAD POF keyword 478
SQLID
definition 549
SQLX edit macro 374
SRTOUT_DATACLASS POF keyword 478
SRTOUT_DATACLASS_ALT POF keyword 478
SRTOUT_EXPDT POF keyword 478
SRTOUT_MGMTCLASS POF keyword 478
SRTOUT_MGMTCLASS_ALT POF keyword 478
SRTOUT_PREFIX POF keyword 478
SRTOUT_PRIQTY POF keyword 479
SRTOUT_RETPD POF keyword 479
SRTOUT_SECQTY POF keyword 479
SRTOUT_STORCLASS POF keyword 479
SRTOUT_STORCLASS_ALT POF keyword 479
SRTOUT_THRESH POF keyword 479
SRTOUT_UNIT POF keyword 479
SRTOUT_UNIT_ALT POF keyword 479
SS command 512
SS symbolic variable 394
SS TEMPLATE descriptor variable 394
SE command 239
SSE ISPF command 44
SSID
creating 162
definition 549
connecting at startup 29
SSID AEXIN keyword 383
SSID OUTPUT descriptor variable
GRPNM symbolic variable 388
JSSID symbolic variable 391
MSSID symbolic variable 392
SS symbolic variable 394
SSID symbolic variable 394
TSSID symbolic variable 395
SSID symbolic variable 394
SSID TEMPLATE descriptor variable
GRPNM symbolic variable 388
JSSID symbolic variable 391
MSSID symbolic variable 392
SSID symbolic variable 394
TSSID symbolic variable 395
ST object type 36
ST symbolic variable 394
START CLONE command 508
START command 268, 508
START DB2 command 201
STARTOVER AEXIN keyword 384
static SQL 370, 373
STATS AEXIN keyword 384
STATS command 511
STATUS command 511
STEP_INCLUDE_MEMBER POF keyword 480
STEP# symbolic variable 394
STEPLIB libraries, setting default values 96
STEPN symbolic variable 394
STEPNAME OUTPUT descriptor variable
STEPN symbolic variable 394
TU1 symbolic variable 395
TU2 symbolic variable 395
TU3 symbolic variable 396
UDOPT symbolic variable 396
WKOWN symbolic variable 397
WKOWNER symbolic variable 397
STEPNAME symbolic variable 394
STEPNAME TEMPLATE descriptor variable
STEPN symbolic variable 394
TU1 symbolic variable 395
TU2 symbolic variable 395
TU3 symbolic variable 396
UDOPT symbolic variable 396
WKOWN symbolic variable 397
WKOWNER symbolic variable 397
STMT installation option 414
STOGROU installation option 414
stogroup object type 35
STOP CLONE command 508
STOP command 268, 508
STOP DB2 command 201
STOPWAIT AEXIN keyword 385
STOPWAIT POF keyword 480
STOPWTSECS AEXIN keyword 385
STOPWTSECS POF keyword 480
storage group object type 178
storage, virtual 66
stored procedure object type 36
stored procedures 506
ADMIN_COMMAND_DB2 160, 165
ADMIN_DS_LIST 160
DSNWZP 160
STOSPACE command 276, 511
STOSPACE utility worklist command 500
string object type 36
STRINGS installation option 414
structure-only baseline
  definition 549
Structured Query Language SQL
  definition 549
SU command 517
SU object type 36, 38
sub-element
  definition 549
SUPPRESS_COMMENTS POF keyword 480
SUTnnndata set 100
SWAP ISPF command 44
switches, CATALOG MANAGER 90
SY command 517
SY object type 36, 174
symbolic variable
  definition 549
symbolic variables
  GDG 109
symbolic variables, list of 386
SYNC
  definition 550
SYNC command 511, 518
sync point
  definition 550
SYNDELETE AEXIN keyword 386
SYNDELETE POF keyword 480
synchronization
  definition 550
SYNLIST AEXIN keyword 386
SYNONYM installation option 414
synonym object type 36
syntax statement conventions 16
SYSCOPY data set, setting JCL options 109
SYSDISC data set 109
SYSERR data set 109
SYSEXEC POF keyword 480
SYSIN in BATCH SEARCH 187
SYSMAPdata set 109
SYSLIB symbolic variable 395
SYSPROC.ADMIN_DS_LIST stored procedure 164
SYSPUNCH data set 109
SYSREC data sets
  LOB 109
  ROWID 109
  setting JCL options 109
SYSTEM command 505
system privilege user object type 36
system privileges for AUTHIDS object type 38
SYSTEM_MLIB POF keyword 480
SYSUID symbolic variable 395
SYSUT data set 109
SYSUT_DATACLASS POF keyword 480
SYSUT_DATACLASS_ALT POF keyword 480
SYSUT_EXPDT POF keyword 481
SYSUT_MGMTCLASS_ALT POF keyword 481
SYSUT_PREFIX POF keyword 481
SYSUT_PRIQTY POF keyword 481
SYSUT_RETPD POF keyword 481
SYSUT_SECQTY POF keyword 481
SYSUT_UNIT POF keyword 482
SYSUT_UNIT_ALT POF keyword 482
SYSUTnnn data set 100
SZDEV T POF keyword 482

T
TA command 330
TABAUTH installation option 414
TABCNST installation option 414
table constraint object type 38
table object type 35, 178
table space mixed object type 38
table space object type 36, 178
table space partition object type 39, 42
table space set object type 39
table space statistics object type 39
table spaces
  dropping 261, 264
  recovering structure and data 268
TABLEPA installation option 414
tables
  CATALOG MANAGER, list of 152
  creating and editing constraints 245
  creating with an existing object 245
  defining columns 245
  for utility support and job generation, list of 152
  identity columns 245
TABLES installation option 414
Tables spaces list object type 39
TABLESH installation option 414
TABLESP installation option 414
TABPRTH installation option 414
TABSTAH installation option 414
TABSTAT installation option 414
TABSTATS command 513
TAILOR command 330, 505
TAPE_EXPDT POF keyword 482
TAPE_RETPD POF keyword 482
TAPE_VOLCNT POF keyword 482
TAPE1 POF keyword 482
TAPE2 POF keyword 482
TAPE3 POF keyword 482
tapes
  setting default processing options 104
  stacking, disabling 104
  using 109
target
definition 550
task ID
definition 550
TASKID AEXIN keyword 386
TB command 518
TB object type 35, 174, 176, 178
TBBR command 216
TBCR symbolic variable 395
TBCRE symbolic variable 395
TBEDIT command 223
TBLPROF installation option 414
TBNAM symbolic variable 395
TBNAM symbolic variable 395
TBNODE symbolic variable 395
TBP command 518
TC command 518
TC object type 38
TDSN option for site profiles 278
TDSNdt 409
template
definition 550
TEMPLATE control statements
  available IBM utilities 292
  creating 291
  including in utility jobs 292
TEMPLATE data set 126
TEMPLATE descriptor variables, list of 386
TEMPLATE_DSN POF keyword 483
temporary work data sets 107
TEMPUNIT POF keyword 483
TERM command 511
THAW command 523
threshold, for alternate unit 109
TI symbolic variable 395
TI TEMPLATE descriptor variable 395
TIME OUTPUT descriptor variable
  HMS symbolic variable 388
  JHMS symbolic variable 389
  TI symbolic variable 395
  TIME symbolic variable 395
TIME symbolic variable 395
TIME TEMPLATE descriptor variable
  HMS symbolic variable 388
  JHMS symbolic variable 389
  TI symbolic variable 395
  TIME symbolic variable 395
TIME4 symbolic variable 395
TIMEPARM POF keyword 483
TIMESTAMP command 505
TM command 518
TM object type 38
TN command 518
TN object type 39, 174
TNCC installation option 409
TNLMR installation option 409
TP command 518
TP object type 39, 42
TR command 518
TR object type 39, 178
traditional list line format 191, 192
TRIGGER installation option 414
trigger object type 39, 178
troubleshooting
authorization to perform SEARCH 178
authorization to use logs 345
availability of actions and object types 338
customizing Primary Menu 331
drop and drop recovery 263
DROP IS switch 264
DROP RECOVERY function 268
Drop Recovery Logs 351
dropping tables 262
DSN1COPY to recover data 268
generating a list 341
initial list filters 341
objects excluded from recovery 268
purging Session Logs 349
recovering changes from logrba 268
recovering incremental image copies 268
recovering indexes 263
referential constraints 262
storage space for dropped tables 262
UNKNOWN COMMAND message 335
WHERE clauses in SEARCH 180, 189
WHERE statements with Session Log Lists 347, 351
wildcard characters in qualifiers 40
TRRS installation option 409
TRTCH POF keyword 483
TRUNCATE command 508
truncation, long names 79
trusted context attribute object type 37
trusted context authorization ID object type 37
trusted context object type 37
Trusted context object type 178
TS command 518
TS object type 36, 174, 178
TS OUTPUT descriptor variable
  IS symbolic variable 389
  IX symbolic variable 389
  IXNAME symbolic variable 389
  IXSPC symbolic variable 389
  SN symbolic variable 394
  SPNAME symbolic variable 394
  TBNAM symbolic variable 395
  TS symbolic variable 395
  TSIX symbolic variable 395
  TSNAME symbolic variable 395
  TS template variable 395
  TS TEMPLATE descriptor variable
  IS symbolic variable 389
  TBNAM symbolic variable 395
  TBNAME symbolic variable 395
  TS symbolic variable 395
  TSNAME symbolic variable 395
  TSCR symbolic variable 395
  TSIX symbolic variable 395
  TSNAM symbolic variable 395
  TSO ID (ectoid) 147
  TSO POFRESET command 505
  TSO POFRESET POF command 505
  TSO region size 29
  TSOPROGRAM POF keyword 483
  TSOSUBEXIT POF keyword 483
  TSS command 518
  TSS object type 39
  TSSID symbolic variable 395
  TT command 518
  TT object type 39, 174
  TU1 symbolic variable 395
  TU2 symbolic variable 395
  TU3 symbolic variable 396
  TYPE OUTPUT descriptor variable
    JOBTYP symbolic variable 390
    LOCREM symbolic variable 391
    LR symbolic variable 391
    OBJT symbolic variable 392
    OBJTYP symbolic variable 392
    PB symbolic variable 393
    PRIBAC symbolic variable 393
    RTYPE symbolic variable 393
    RUNITYP symbolic variable 394
    TYPE symbolic variable 396
  TYPE parameter 149
  TYPE symbolic variable 396
  TYPES command 519

U

U line command 523
UA command 518
UA object type 39
UCMD symbolic variable 396
UCOMD 409
UCOMD installation option 139
UDOPT symbolic variable 396
UID symbolic variable 396
ULLQ POF keyword 483
ULLQ symbolic variable 396
creating 127
updating directly 129
updating in options panels 129
using multiple 130
user privileges, copying by privilege type 313
user profile data set for user profiles 278
user variables 127
USER_HLQ POF keyword 491
USER_VAR1_CHAR POF keyword 491
USER_VAR2_CHAR POF keyword 491
USER_VAR3_CHAR 491
USER_VAR4_CHAR POF keyword 491
USER_VAR5_CHAR POF keyword 491
user-written commands
&CLIST parameter 143
&DB2MAX parameter 143
&DB2MIN parameter 143
&HELP parameter 143
&LOAD parameter 143
&LOG parameter 143
&LSTO parameter 143
&NLIST parameter 143
&NOSERVER parameter 143
&NUM parameter 143
&OBJECTS parameter 143
&PAREN parameter 143
&PAREN parameter 143
&PLAN parameter 143
&WFEK parameter 143
commands table 138
creating 139
object types 149
writing as CLIST 141
writing as program 141
USER1 symbolic variable 396
USER2 symbolic variable 396
USERAUT installation option 414
USERID OUTPUT descriptor variable
USERID symbolic variable 396
ZACCTNUM symbolic variable 398
ZPREFIX symbolic variable 398
ZUSER symbolic variable 398
USERID symbolic variable 396
USERNAM installation option 414
USRCOMND member 139, 147
UT symbolic variable 396
UT TEMPLATE descriptor variable
ALID symbolic variable 387
FCMD symbolic variable 388
JQID symbolic variable 390
UCMD symbolic variable 396
USER1 symbolic variable 396
USER2 symbolic variable 396
UT symbolic variable 396
UTID symbolic variable 396
UTIL symbolic variable 396
UTILID symbolic variable 396
UTILPFX symbolic variable 397
UTILSFX symbolic variable 397
WKID symbolic variable 397
WORKID symbolic variable 397
WORKID8 symbolic variable 398
UTID OUTPUT descriptor variable 396
UTID symbolic variable 396
UTIL command 512
UTIL OUTPUT descriptor variable
ALID symbolic variable 387
FCMD symbolic variable 388
JQID symbolic variable 390
LI symbolic variable 391
LIST symbolic variable 391
OBJT symbolic variable 392
OBJTYP symbolic variable 392
UCMD symbolic variable 396
USER1 symbolic variable 396
USER2 symbolic variable 396
UT symbolic variable 396
UTID symbolic variable 396
UTIL symbolic variable 396
UTILID symbolic variable 396
UTILPFX symbolic variable 397
UTILSFX symbolic variable 397
WKID symbolic variable 397
WORKID symbolic variable 397
WORKID8 symbolic variable 398
UTIL profileID command 512
UTIL symbolic variable 396
UTILID symbolic variable 396
UTILITY command 286, 512
utility commands, list of 508
utility edit propagation 289
utility module names, setting default processing options 119
utility processing
BMC Software utilities 275
IBM DB2 utilities 276
JCL data set name 281
JCL options 281
multiple utilities 286
number of control statements allowed 278
single utility 281
User Profile data set name 281
Utility ID variables 281
utility profile data set
allocating 279
utility profile data set, creating 279
UTILITY profile ID command 512
utility profiles
allocating a data set 278
changing options values 289
creating from an existing profile 287
editing 289
last-used profile ID 290
online tutorial 278
PROFILE command 291
profile ID 281
setting up 278
site profiles 278
User Profile data set name 281
user profiles 278
UTILITYID AEXIN keyword 386
UTILPFX symbolic variable 397
UTILSFX symbolic variable 397
UVR1 symbolic variable 397
UVR2 symbolic variable 397
UVR3 symbolic variable 397
UVR4 symbolic variable 397
UVR5 symbolic variable 397

V
V line command 523
VAR command 518
VAR installation option 410
VAR object type 39
VARAUTH installation option 410
variable
definition 552
variables object type 39
VCAT AEXIN keyword 386
VCAT allocation
definition 552
VCAT symbolic variable 397
version information, displaying 61
versioning
definition 552
view object type 35, 178
VIEW table 152
VIEWDEP installation option 414
VIEWS installation option 414
virtual storage 66
VL command 518
VL object type 39
volume object type 39
VOLUMES installation option 414
VSAM object
data set sizing 100
JCL generation 356
sampling 356
VW command 518
VW object type 35, 174, 176, 178

W
Wait-for-Enter commands 55
WARNRC AEXIN keyword 386
WDSN installation option 410
WHERE clauses in SEARCH 207
where command is executed parameter (exccsrc) 147
wildcard
definition 552
wildcard characters
cascading authorizations 304
fixed-length CHAR columns 41
in host variables 185
in quick-search 188
in WHERE clause 180
two-part object names 41
use in Qualifier field 40
with saved SEARCH variables 188
WKID symbolic variable 397
WKOWN symbolic variable 397
WKOWNER symbolic variable 397
work data sets
permanent 109
temporary 107
used by utilities 100
work ID
definition 552
work ID name
definition 552
work ID name template
definition 553
work ID owner
definition 553
WORK_DATACLASS POF keyword 491
WORK_MGMTCLASS POF keyword 491
WORK_STORCLASS POF keyword 491
WORKID AEXIN keyword 386
WORKID symbolic variable 397
WORKID8 symbolic variable 398
worklist
definition 553
worklist commands
-BMCU (BMCU Execute a BMC Utility) 495
-DEBUG  (Debug) 495
-SQLM (SQL Statement) 499
worklist execution log
definition 553
worklist parallelism
definition 553
WRKnnn data set 100

X
X ALL command 177
X command 505
X line command 177
X line designator 55
XC command 518
XC object type 39
XIM
definition 553
XML relationship object type 39, 178
XML string object type 39
XML strings object type 178
XMLREL installation option 414
XMLSTR installation option 414
Xnnn command 505
XODSN installation option 411
XR command 519

Y
YE symbolic variable 398
YE TEMPLATE descriptor variable 398
YEAR OUTPUT descriptor variable 398
YEAR symbolic variable 398
YMD symbolic variable 398
YY OUTPUT descriptor variable 398
YY symbolic variable 398
YYDDD symbolic variable 398
YYYYDDD symbolic variable 398

Z
Z line command 523
ZACCTNUM SLIB variable 398
ZACCTNUM symbolic variable 398
ZOOM command 244, 523
ZOOM ISPF command 44
ZPREFIX SLIB variable
  PREFIX symbolic variable 393
  ZPREFIX symbolic variable 398
ZPREFIX symbolic variable 398
ZSYSID SLIB variable 398
ZSYSID symbolic variable 398
ZUSER SLIB variable
  UID symbolic variable 396
  USERID symbolic variable 396
  ZUSER symbolic variable 398
  ZUSER symbolic variable 398

CATALOG MANAGER for DB2 User Guide